jamk

AI-Driven Patient Recruitment for Clinical Trials Using NLP on E-lääkärinlausunto in Finland's Healthcare System

Waseem Pasha

Master's thesis
February 2025
Master's Degree Programme in Artificial Intelligence and Data Analytics

jamk | Jyväskylän ammattikorkeakoulu
University of Applied Sciences

Waseem Pasha

AI-Driven Patient Recruitment for Clinical Trials Using NLP on E-lääkärinlausunto in Finland's Healthcare System

Jyväskylä: Jamk University of Applied Sciences, February 2025, 40 pages.

Master's Degree Programme in Artificial Intelligence and Data Analytics

Permission for open access publication: Yes

Language of publication: English

Abstract

Traditional drug testing through clinical trials is time-consuming, expensive, and often limited in scope, prompting the need for more efficient methods. This study explores an AI-driven framework leveraging Natural Language Processing (NLP) on Finland's medical certificates—commonly referred to as E-lääkärinlausunto—to streamline patient recruitment. These documents, include A, B, C, and E certificates in Finnish Health care system, but our focus will mainly be on E-certificates as this contain critical healthcare data such as diagnoses, treatments, and patient histories. By employing Optical Character Recognition (OCR) for data extraction and NLP techniques—Named Entity Recognition, ICD mapping, temporal analysis, and keyword matching—unstructured medical text is transformed into structured formats suitable for automated eligibility checks.

The resulting patient profiles are compared against predefined clinical trial criteria, enabling a high-precision matching process that reduces recruitment time and cost. A feedback loop further refines accuracy, as clinicians validate or reject suggested matches. and can scale securely on platforms complying with GDPR and HIPAA regulations.

This approach not only accelerates clinical trial timelines but also enhances inclusivity by identifying underrepresented groups more effectively. Ultimately, AI-driven patient recruitment holds the potential to revolutionize drug testing, enabling faster, safer, and more diverse trials that benefit healthcare providers, pharmaceutical companies, and patients alike.

Building on these findings, this study will further investigate how advanced NLP methods applied to Finland's diverse medical certificates can reshape the clinical trial process. By systematically harnessing the granular patient data contained in E-lääkärinlausunto, it aims to demonstrate a scalable approach for identifying trial candidates more quickly and accurately. This direction not only has the potential to reduce the overall costs and timelines associated with traditional drug testing but also fosters inclusivity by capturing underrepresented patient demographics. Ultimately, the research aspires to illustrate how AI-driven patient recruitment can accelerate the discovery of safer, more effective treatments and usher in a future of truly personalized medicine.

Keywords/tags (subjects)

Clinical Trials, Drug Testing, Patient Recruitment, Patients Information,

Miscellaneous (Confidential information)

N/A

Contents

Figures

1 Introduction

"If you think research is expensive, try disease."

— Mary Lasker, American health activist and philanthropist

The release of any medicine into the market demands rigorous evaluation through clinical trials. These studies are fundamental to modern healthcare, functioning as the gatekeepers of safety, efficacy, and quality assurance for new drugs and medical interventions (Topol, 2019). Despite their indispensable role, clinical trials often face systemic inefficiencies—particularly in participant recruitment—that inflate costs and impede timely advancement of life-saving therapies (Getz & Campo, 2018). Industry data shows that **80%** of trials fail to enroll on schedule, sometimes accruing up to **$8 million per day** in delayed drug approvals (Chopra et al., 2023). Such setbacks translate into prolonged patient suffering, missed profit windows for pharmaceutical companies, and broader public health consequences when critical treatments do not reach the market swiftly (Liu et al., 2021).

In addition to these financial and temporal pressures, concerns about **trial diversity** exacerbate the recruitment crisis. Take, for instance, the fact that less than 5% of participants in cancer trials are non-white? This significant gap not only affects the reliability of the results but also continues to fuel healthcare inequalities (Carlson et al., 2021). When recruitment is biased or incomplete, it can lead to a lack of effectiveness for drugs in real-world situations, especially when certain groups with distinct genetic traits or health conditions are overlooked (Buolamwini & Gebru, 2018). To tackle these issues, Artificial Intelligence (AI) is starting to transform the clinical research field, offering automated data extraction, sophisticated analytics, and streamlined workflows (Esteva et al., 2017; Liu et al., 2021). Various deep learning and natural language processing (NLP) techniques—like Optical Character Recognition (OCR), Named Entity Recognition (NER), and predictive analytics—have already shown remarkable success in tasks such as medical image classification and pathology detection (Zhavoronkov et al., 2019; Topol, 2019).

Yet, despite the global enthusiasm for AI, each healthcare system presents distinctive data formats, legal frameworks, and cultural contexts. In Finland, a country known for its advanced digital infrastructure, clinical trials still encounter recruitment hurdles—particularly in handling **E-lääkärinlausunto**, an electronic medical certificate system storing valuable patient data in image-

based or unstructured formats (Kanta Services, 2023). These documents are bilingual (Finnish/Swedish), often contain handwritten components, and are spread across multiple hospital districts, creating “data silos” that hamper centralized analysis (FinRegistry, 2023). The General Data Protection Regulation (GDPR) and Finland’s Act on the Secondary Use of Health and Social Data (552/2019) further underscore the need to handle sensitive patient information responsibly (Ministry of Social Affairs and Health, 2019). Consequently, bridging advanced AI algorithms with the complexities of Finnish healthcare requires not just robust technical solutions but also compliance with strict data privacy laws and ethical considerations (Buolamwini & Gebru, 2018; Hassan et al., 2023).

Against this backdrop, the current study proposes leveraging **Natural Language Processing (NLP)**—encompassing OCR, Named Entity Recognition, ICD-10 mapping, and temporal analysis—to **automate patient recruitment** in Finland. By transforming E-lääkärinlausunto’s unstructured narratives into structured data, trial coordinators can quickly assess patient eligibility for complex protocols, reduce manual overhead, and potentially cut recruitment time by up to **50%** (Chopra et al., 2023; Chow et al., 2023). Beyond cost savings, these systems can really help foster inclusivity by pinpointing underrepresented groups that might otherwise be overlooked in manual screening processes (Carlson et al., 2021). This study thus aims to validate the hypothesis that AI-driven approaches not only streamline recruitment logistics but also encourage broader patient participation, aligning scientific progress with equitable healthcare access.

In the upcoming sections, this study will (1) identify the main challenges in Finnish clinical trial recruitment, (2) describe the range of AI techniques used, such as transformer-based named entity recognition and advanced optical character recognition, and (3) evaluate how well these systems perform in both synthetic and real E-lääkärinlausunto scenarios. Through these discussions, it emphasizes how Finland’s strong digital infrastructure, paired with emerging AI technologies, can act as a global model for ethical and efficient management of clinical trials—ultimately speeding up the development of therapies that can benefit a wide array of populations (Ministry of Social Affairs and Health, 2019; Klassen, 2016)

Before any medication can be released to the market, it must undergo rigorous testing and approvals, a process which is known as clinical trials in the medical field. These trials are a form of research focused on discovering new tests and treatments while assessing their impact on human

health. Patients participate in these studies to help evaluate medical procedures, which may include drugs, biological products such as cells, surgical or radiological procedures, medical devices, behavioural therapies, and preventive care measures.

Before clinical trials can begin, they must be carefully planned, reviewed, and approved to ensure safety and reliability. These studies are open to people across all age groups, including children, who may qualify to take part in testing these new medical advancements.

The process of developing and approving new medical treatments—a journey that links scientific innovation with the health of individuals—relies on the successful execution of clinical trials. These trials are fundamental to evidence-based medicine, serving as the ultimate decision-makers regarding the safety, effectiveness, and therapeutic benefits of drugs, devices, and interventions. Yet, despite their vital role, clinical trials are often hindered by systemic inefficiencies that can delay breakthroughs, increase costs, and exclude essential patient populations. This difficulty is most evident during the participant recruiting phase, which is so filled with challenges that it has emerged as the main bottleneck in modern clinical research. In a time characterized by artificial intelligence (AI) and data-driven healthcare, this study proposes a transformative idea: utilizing Natural Language Processing (NLP) on Finland's electronic medical certificates, known as E-lääkärinlausunto, to revolutionize patient recruitment for clinical trials. By converting unstructured medical narratives into actionable insights, this research aims to bridge a significant gap in Finland's healthcare innovation ecosystem while offering a model for global adoption.

The current process of testing drugs through clinical trials, which is considered the gold standard, faces critical challenges such as high costs, long durations, and limited effectiveness. Modern clinical trials typically involve selecting locations, recruiting participants, administering the drug, and monitoring the outcomes. However, this trial-and-error method is becoming increasingly expensive and time-consuming, often taking over five years and costing billions.

AI presents a transformative solution to these challenges. For example, during the COVID-19 pandemic, AI tools like Delphi were employed to predict optimal trial locations months in advance. This innovative approach not only accelerated trial timelines by eight weeks but also reduced participant requirements by 25%, enhanced trial diversity, and provided efficacy data on variants. Beyond expediting trials, AI can improve accessibility for underrepresented groups, simplify participation, and personalize treatments to individual physiologies.

These advancements highlight the potential of AI to revolutionize drug testing, making healthcare more efficient, inclusive, and effective, ultimately contributing to better, longer, and healthier lives.

1.1 Importance Of Clinical Trials

Clinical trials play a crucial role in advancing medical science. Before any treatment can be made available to the public, it has to go through a thorough evaluation process categorized into four phases, safety testing (Phase I), efficacy assessment (Phase II), large-scale validation (Phase III), and post-market studies (Phase IV) (Umscheid et al., 2011).

Clinical trial statistics show that there are some real challenges when it comes to recruiting patients, and these challenges can significantly affect the efficiency and cost-effectiveness of medical research. Shockingly, around 80% of clinical trials are either delayed or canceled because of recruitment issues. Specifically, 37% of trial sites are unable to enroll enough volunteers, and 11% don't manage to bring in a single participant. These recruitment struggles lead to major financial implications, as clinical trials account for nearly 40% of the U.S. pharmaceutical research budget, which adds up to around $7 billion each year. Interestingly, patient recruitment itself makes up 40% of that budget, which is about $1.89 billion. When clinical trials are delayed, sponsors can face costs ranging from $600,000 to $8 million for each day that the development and launch of a product are postponed (McDowell, 2013).

1.2 The Recruitment Bottleneck: Why Traditional Methods Fail

Recruiting patients for clinical trials using traditional methods often falls short because of various logistical, informational, and perceptual hurdles. Pharmaceutical companies, despite being skilled in commercial marketing, often don't apply those same strategies when it comes to sharing the value of clinical trials with investigators and patients. As a result, many potential participants either remain in the dark about these trials or have misconceptions about their purpose. On top of that, strict inclusion and exclusion criteria can drastically limit the number of eligible participants, turning what could be promising groups into recruitment challenges. Physicians also contribute to this problem—many are either unaware of the trials that exist or wrongly assume their patients can't participate, which means missed opportunities for referrals (Frank, 2004).

In Finland, the situation is further complicated by the fragmented system of E-lääkärinlausunto, which are electronic medical certificates that compile diagnoses, treatments, and patient histories from both primary and specialty care. These documents are often saved as PDFs, TIFFs, or JPEGs and contain a wealth of unstructured text in Finnish, Swedish, or bilingual formats. Unfortunately, their variety—different templates, handwritten notes, and non-standard terminology—makes them difficult for automated analysis to handle.

Consider a typical scenario: A Helsinki-based researcher seeking patients with type 2 diabetes (ICD-10: E11) for a Phase III trial must manually review thousands of E-lääkärinlausunto certificates to identify candidates meeting criteria such as HbA1c > 7%, BMI ≥ 30, and no renal impairment. This process, which can take **6–12 months**, is further hampered by language-specific challenges, including:

Finnish medical jargon (e.g., “diabetes tyyppi 2” vs. “type 2 diabetes”).

Temporal ambiguities (e.g., “diagnosed in 2021” vs. “history of diabetes since childhood”).

Data silos between Finland’s 21 hospital districts.

The result? Missed eligibility, delayed trials, and a healthcare system struggling to translate research into practice.

In Figure 1 below as you can see second stage of Drug testing is Recruitment, this is the most crucial and important stage, many people are needed for testing the drugs as it is tested in 4 phases, this is our focus area in this study, because many people are needed in all 4 phases, recruitment is difficult.

It is also very important to recruit people from different background, having diversity in recruitment is important because people may experience the same disease differently. It’s essential that clinical trials include people with a variety of lived experiences and living conditions, as well as characteristics like race and ethnicity, age, sex, and sexual orientation, so that all communities benefit from scientific advances

1.3 Stages of Clinical Trials

There are basically 4 stages in testing a drug, this has been shown in Figure 1 below

Figure 1: Stages of drug testing

1. **Location**: deciding location this basically depends on Pharma company where it is located and to which market it serves and stage 2 involves
2. **Recruitment**: Many people are needed for testing the drugs as it is tested in 4 phases, this is our focus area in this study, because many people are needed in all 4 phases, recruitment is difficult.
3. **Monitoring**: Once the Drug is released to market monitor the performance of it over time.
4. **Analysis**: Keep collecting data to analyze the efficiency of your drug and keep comparing it with the other drugs in market to see effectivity over time

1.4 Problem Statement

1.4.1 Need for AI-Driven Recruitment in Clinical Trials

Patient recruitment for clinical trials is a time-consuming and expensive process, often relying on manual screening of medical records to determine eligibility. In Finland, the E-lääkärinlausunto (electronic medical certificate) contains unstructured text-based medical data, which presents a challenge for efficient patient identification. The lack of automation in processing these documents results in delays, higher costs, and potential loss of eligible candidates.

AI-driven Natural Language Processing (NLP) can transform this unstructured medical text into structured data, enabling automated patient recruitment. However, the effectiveness of various AI

methods in this process remains underexplored. This study investigates what specific AI techniques—including Optical Character Recognition (OCR), Named Entity Recognition (NER), ICD code mapping, and Temporal Analysis—can be applied to extract relevant patient data from E-lääkärinlausunto documents and streamline patient recruitment for clinical trials in Finland.

1.5 Research Question

1.5.1 What AI-driven methods can optimize patient recruitment for clinical trials using E-lääkärinlausunto in Finland?

Research Question: *How can advanced AI-based techniques—encompassing Optical Character Recognition, Named Entity Recognition, ICD-10 mapping, temporal analysis, and keyword matching—be effectively leveraged to transform E-lääkärinlausunto into actionable data for patient recruitment in Finland's clinical trials?*

Rationale and Significance

Finding patients for clinical trials is often one of the most resource-intensive tasks, which can lead to delays and increased costs (Chopra et al., 2023). In Finland, the process of determining eligibility using E-lääkärinlausunto—unstructured or scanned medical certificates—remains largely manual and inefficient (Klassen, 2016). This not only slows down trial timelines but also risks leaving out underrepresented groups (Chow et al., 2023).

Why This Matters

Making the most of E-lääkärinlausunto documents can significantly boost trial accessibility and accelerate medical research. OCR tools like Tesseract are great for converting scanned records into text, while NER models such as spaCy or BioBERT focus on extracting crucial clinical data. By using ICD-10 mapping, we can standardize diagnoses, which helps in efficiently matching patients to trials. Temporal analysis is key for assessing eligibility based on when symptoms appear, and keyword matching is essential for capturing specific terms that might slip past more general models (Hassan et al., 2023).

Expected Impact

Recent meta-analyses report that AI-driven methods have the capacity to cut recruitment periods

by 50% and costs by nearly 30%, all while improving demographic diversity and trial accuracy (Chopra et al., 2023). Implementing these end-to-end AI techniques in the Finnish healthcare ecosystem not only alleviates administrative burdens but also democratizes access to cutting-edge therapies. Faster and more accurate participant matching means that life-saving interventions can reach markets more swiftly, and patients—particularly those in underserved groups—can benefit from earlier, more targeted treatments (Chow et al., 2023). As Finland is known for its robust digital infrastructure and strict data governance, deploying AI-driven recruitment solutions can serve as a global blueprint, demonstrating how to merge efficiency with ethical and privacy considerations (Ministry of Social Affairs and Health, 2019).

1.5.2 How can integrating advanced NLP techniques further improve recruitment accuracy and reduce processing time?

Research Question:

In what ways do advanced natural language processing (NLP) approaches—such as transformer-based Named Entity Recognition (NER), domain-adapted language models, and semantically enriched information extraction—enhance the speed and precision of patient recruitment from E-lääkärinlausunto beyond conventional AI methods?

Rationale and Significance

While fundamental AI-driven approaches like Optical Character Recognition (OCR) and simple keyword matching already demonstrate benefits in automating patient recruitment, especially for unstructured Finnish medical certificates (Chopra et al., 2023), the degree of accuracy and speed can still vary greatly. Complex scenarios—such as detecting nuanced diagnoses, interpreting time-bound conditions, and parsing bilingual or dialect-specific content—often necessitate more sophisticated NLP solutions. For example, an E-lääkärinlausunto may mention “krooninen keuhkosairaus” (chronic lung disease) in a footnote or reference an incidental finding in Swedish, requiring deeper semantic analysis. Traditional rule-based or even basic machine-learning approaches can miss these subtleties, leading to under- or over-inclusion of candidates (Klassen, 2016).

Why This Matters

Clinical trials routinely handle large volumes of electronic physician statements containing critical data points—diagnostic codes, treatment timelines, and potential comorbidities. Simply converting text to digital form and matching keywords does not fully address complexities like negations (e.g., "No evidence of pneumonia"), ambiguous temporal indicators (e.g., "Symptoms worsened over several months"), or domain-specific abbreviations. Advanced NLP models, such as transformer-based frameworks (e.g., BioBERT, FinBERT), can interpret longer contexts and capture linguistic nuances across multiple sentences (Chow et al., 2023). This ability significantly reduces false positives and negatives, thereby refining recruitment accuracy. Additionally, NER techniques augmented with contextual embeddings can handle morphological complexity in Finnish and Swedish, differentiating "sydän" (heart) from "sydänkohtaus" (heart attack), for instance, thus ensuring more precise extraction of potential eligibility factors (Hassan et al., 2023).

Expected Impact

By integrating these advanced NLP techniques, recruitment pipelines stand to gain in both speed—through automated data extraction and reduced human verification—and accuracy—by correctly filtering candidates who genuinely meet trial criteria. Studies show that advanced NLP can boost precision by as much as 10–15% over baseline machine-learning methods in high-stakes medical data scenarios (Chopra et al., 2023). This translates into more efficient clinical trials, as recruitment phases often consume the bulk of total trial time and budget. Enhanced accuracy also increases the likelihood of including diverse patient populations, mitigating longstanding biases. Ultimately, the fusion of advanced NLP with existing AI tools lays the groundwork for a robust, adaptive system that identifies eligible participants in real time, thereby aligning with Finland's push toward more agile, data-driven healthcare (Ministry of Social Affairs and Health, 2019).

1.5.3 What are the impacts of AI-driven recruitment on trial diversity and inclusivity?

Research Question

How does deploying AI-driven patient recruitment systems—especially those leveraging E-lääkärinlausunto—affect the representation of underrepresented groups and overall inclusivity in clinical trials conducted in Finland?

Rationale and Significance

Traditional clinical trial enrollment processes often struggle with skewed demographics, resulting in trials that underrepresent certain ethnic groups, rural populations, the elderly, or individuals with complex comorbidities (Chopra et al., 2023). This lack of diversity undermines the external validity of clinical findings and may unintentionally perpetuate healthcare disparities. In Finland, E-lääkärinlausunto consolidates a wide range of patient data—spanning multiple demographics and geographical locations—yet manual reviews can miss or deprioritize rarer subpopulations (Klassen, 2016). By contrast, AI-driven tools provide a systematic, rules-based approach to scanning and shortlisting patient candidates, potentially uncovering eligible participants who might otherwise be overlooked due to language barriers, complex conditions, or limited healthcare access (Chow et al., 2023).

Why This Matters

Ensuring diversity in clinical trials isn't just a matter of ethics; it's vital for achieving strong clinical outcomes. When we exclude key patient groups—like older adults or specific language communities the data we collect may not accurately reflect real-world clinical scenarios (Hassan et al., 2023). In Finland, where E-lääkärinlausunto documents can be challenging for manual recruiters to interpret, AI-driven recruitment systems can effectively target patients with rare diagnoses or those from underserved areas. Additionally, features like advanced Named Entity Recognition (NER) and ICD mapping can help pinpoint potential candidates with less common or multiple health conditions, enhancing the chances that trials will include a diverse cross-section of Finnish society (Klassen, 2016).

Expected Impact

By adopting these automated screening strategies, healthcare organizations can better balance demographics in trial enrolment—taking into account both condition prevalence and social determinants of health. This can lead to more equitable access to state-of-the-art treatments and produce richer clinical data that reflects the diversity of Finland’s population (Chopra et al., 2023). Studies suggest that AI-assisted recruitment can lower both time and costs while enhancing representation among rural residents, linguistic minorities, and older adults. Consequently, having a more inclusive pool of participants helps to identify differences in drug efficacy across various demographics, which in turn supports the development of therapies that are effective across different populations (Hassan et al., 2023).

Ultimately, this question addresses how AI can help balance efficiency with equitable healthcare—a vital concern in Finland's ongoing digital transformation and a model for global research ecosystems (Ministry of Social Affairs and Health, 2019).

1.5.4 How do ethical considerations and data privacy concerns influence the implementation and acceptance of AI in clinical trial recruitment?

Research Question

In what ways do ethical frameworks, regulatory mandates (e.g., GDPR, Finnish data protection laws), and stakeholder trust issues shape the deployment, efficacy, and social acceptance of AI-driven recruitment systems for clinical trials using E-lääkärinlausunto?

Rationale and Significance

Although AI-driven recruitment offers the promise of faster, more cost-effective clinical trials, it simultaneously raises critical ethical and data privacy questions. In Finland, the E-lääkärinlausunto is filled with rich patient information—often including sensitive diagnoses, treatments, and personal identifiers. When we use advanced machine learning and natural language processing (NLP) to sift through these records, there's a risk of unintentionally exposing patient data to unauthorized access or misuse if we're not careful (Klassen, 2016). Plus, the biases that can creep into AI algorithms—often due to training datasets that don't represent everyone—might either overlook certain demographic groups or unfairly flag others, which can worsen healthcare inequalities (Hassan et al., 2023).

Why This Matters

The success of AI-driven solutions in clinical research depends not only on their technical strength but also on the level of public trust and commitment to ethical practices. Under the European Union's General Data Protection Regulation (GDPR) and Finland's Act on the Secondary Use of Health and Social Data (552/2019), organizations are obligated to prioritize data minimization, secure informed consent, and ensure transparent processing (Ministry of Social Affairs and Health, 2019). Any breach of confidentiality or perceived infringement on patient rights could seriously damage trust in the entire AI approach, leading to lawsuits, the loss of ethical approvals, or reluctance from clinicians and patients to participate in future studies (Chow et al., 2023).

Expected Impact

Addressing ethical and privacy concerns is key to how quickly AI-driven recruitment is adopted and how it's perceived by society. Techniques like pseudonymization, federated learning, and differential privacy help reduce the risk of re-identification while still enabling effective data analysis (Chopra et al., 2023). By implementing transparent audit trails and explainable AI algorithms, we can further reduce skepticism, reassuring stakeholders that the criteria for patient selection are fair and based on solid science (Hassan et al., 2023). If we manage these challenges well, these protective measures can lead to greater acceptance, allowing research institutions, regulators, and the public to see AI as a trustworthy ally in speeding up clinical trials. Therefore, understanding how ethical and privacy factors interact with technology deployment is essential for establishing operational limits and ensuring the long-term viability of AI in Finnish healthcare (Klassen, 2016).

1.6 Scope and Limitations

1.6.1 Scope

In this study, we're looking at how AI-driven Natural Language Processing (NLP) techniques can be used to improve patient recruitment for clinical trials, particularly through E-lääkärinlausunto, Finland's electronic medical certificates. The research encompasses everything from document preprocessing and data extraction to the automated matching of patients with trials, utilizing AI methods like Optical Character Recognition (OCR), Named Entity Recognition (NER), ICD-10 Code Mapping, and Temporal Analysis. The scope extends to evaluating AI's efficiency compared to manual recruitment, examining processing speed, accuracy, and inclusivity. The study aligns with Finland's GDPR and healthcare regulations, ensuring ethical AI implementation. The findings aim to streamline recruitment, reduce costs, and improve patient diversity in trials.

1.6.2 Limitations

Despite AI's potential, data privacy and security concerns remain critical limitations. E-lääkärinlausunto contains sensitive patient data, requiring strict compliance with GDPR. Additionally, OCR accuracy varies with document quality and formatting, affecting data extraction reliability. NER models struggle with Finnish-language medical terms, necessitating extensive training on localized datasets. Another limitation is limited access to real-world medical records, which may restrict AI model validation. Moreover, AI-driven patient selection may introduce biases if training datasets

lack diversity. The study also does not address physician and patient acceptance of AI-based recruitment, a key factor in real-world implementation. While AI can reduce recruitment time by 50%, human oversight remains necessary to ensure ethical decision-making.

2 Literature Review

2.1 Clinical Trial Recruitment: Traditional Approaches

This chapter is the main focus area of this study

Figure 2: Recruitment phases

Patient recruitment strategies differ across the four phases of clinical trials, each bringing its own set of challenges and goals. In **Phase I** trials, which usually involve a small group of healthy volunteers or patients, the main focus is on evaluating safety, dosage, and side effects. Here, recruitment hinges on the willingness of volunteers to take part, even with the risks being somewhat unknown. Often, participants are motivated by compensation and the promise of close monitoring. However, due to strict inclusion criteria and safety concerns, finding volunteers can be quite tough and selective (Umscheid, Margolis, & Grossman, 2011).

As we move to **Phase II** trials, the participant pool widens to evaluate how effective the treatment is while still keeping an eye on safety. Recruitment in this phase becomes more specific, targeting patients who have the condition being studied. With tighter eligibility criteria, it can be a challenge to find the right candidates. **Phase III** trials, which involve large-scale comparisons with standard

treatments, require a broad and inclusive recruitment strategy to ensure that the general population is represented, which is crucial for external validity. This phase often faces the most delays due to the need for a large sample size. Finally, **Phase IV** trials take place after a treatment has been approved, focusing on long-term effects and real-world effectiveness. Recruitment tends to be easier here since the intervention is already deemed safe and may be part of routine care, but it still requires robust outreach to encourage ongoing participation in monitoring (Umscheid et al., 2011).

2.2 Challenges of Traditional Recruitment in Clinical Trials

Recruitment and retention continue to be significant challenges in clinical trials, often leading to longer study durations, rising costs, and reduced statistical power. In a qualitative study, clinical research professionals from the UK identified four main categories of barriers: subject-related, investigator-related, protocol-related, and systemic factors. They reported that subject-related barriers include unrealistic expectations, language and cultural differences, and logistical issues like relocation. Investigator-related challenges included time constraints, low motivation, and inadequate feasibility assessments. They also pointed out that complex study designs and tight schedules create protocol-related hurdles, while delays in ethical approvals and poor site selection add to systemic challenges (Sullivan, J. 2004).

Kadam et al. (2016) highlighted the following obstacles to recruitment: study protocols were too complicated (38%), lack of awareness concerning the clinical trial was prevalent (37%), and there were cultural concerns related to participation (37%). Along with these factors, patients mentioned not wanting to participate due to fears concerning possible adverse events, distrust fueled by the media, and logistical issues like long-distance travel.

More strides are being made in the field of precision oncology, but researchers still struggle to inclusively engage a diverse population in clinical trials. As for the participants, researchers have tended to recruit younger, healthier, and mostly white individuals from high-income areas. As many participants as possible were enrolled, but as a result, older adults, rural residents, racial and ethnic minorities, people with chronic illnesses, and even immunocompromised people were overlooked. Furthermore, trial sponsors and regulatory agencies impose strict logistical requirements and eligibility criteria. These include: long distances to travel, limited languages available,

and no digital access, which makes them ineligible on top of being marginalized. All of these structural obstacles worsen the inequity in access to treatment and trials which lowers the fairness of clinical research (Vidal et al., 2024).

The research ecosystem's structural and cultural barriers impose an absolute constraint on the inclusive and efficient recruitment of participants for clinical trials. Underrepresentation of several minority groups, overly complex processes for informed consents, fragmented technological infrastructure, and lack of trust from the community are some of the factors that contribute to low participation rates as highlighted by The National Academies In addition, fragmented clinical research as a routine medical practice, lack of workforce diversity, and the digital divide all perpetuate these barriers. These issues highlight the need for immediate development of decentralized approaches that are more focused on the patients and use modern technologies, such as AI and NLP, to automate screening for eligibility and outreach early in the recruitment process targeting underserved populations. As emphasized in the Vision 2030 report, culturally and technologically tailored innovations are required to address these issues and fundamentally transform the accessibility proportions for clinical trials (National Academies of Sciences, Engineering, and Medicine, 2022).

2.3 AI in Healthcare: An Overview

Artificial Intelligence (AI) is rapidly transforming healthcare by enhancing diagnostic accuracy, streamlining clinical workflows, and supporting preventive care. From robotic surgeries to medical imaging, AI is being utilized in a variety of areas, including clinical decision support, patient monitoring, and optimizing clinical trials. These advanced systems harness the power of machine learning (ML), deep learning (DL), natural language processing (NLP), and machine vision to make sense of intricate medical data, automate routine tasks, and enhance personalized patient care. Remarkably, AI-driven platforms have demonstrated significant improvements in diagnostic accuracy and treatment results, all while cutting down on healthcare costs and easing the burden on providers. With a projected global savings potential of $150 billion by 2026, the integration of AI is seen as both a crucial technological advancement and a promising economic opportunity in today’s healthcare landscape (Väänänen et al., 2021

One of the most significant ways AI is making a difference is in diagnostics. Machine learning algorithms, which are trained on extensive datasets that include medical imaging, genomic information, and electronic health records (EHRs), can spot anomalies with a level of accuracy that matches or even exceeds that of human experts. Take, for instance, Google's DeepMind, which created an AI system capable of diagnosing retinal diseases from optical coherence tomography (OCT) scans with an impressive 94% accuracy. This advancement helps to cut down on diagnostic delays for conditions like diabetic retinopathy (Esteva et al., 2017). In a similar vein, AI-driven tools like PathAI are transforming the field of pathology by examining biopsy slides to pinpoint cancerous tissues, making life easier for oncologists. And it doesn't stop there; natural language processing (NLP) models are also pulling valuable insights from unstructured clinical notes, such as physician narratives in Finland's E-lääkärinlausunto, which allows for automated coding of diagnoses (like ICD-10) and treatments.

AI also drives innovation in drug discovery and development, a traditionally slow and costly process. ML algorithms predict molecular interactions, screen potential drug candidates, and optimize clinical trial designs. For instance, Insilico Medicine used generative AI to identify a novel drug target for fibrosis in just 18 months—a process that typically takes years (Zhavoronkov et al., 2019). In clinical trials, AI platforms like Deep 6 AI accelerate patient recruitment by mining EHRs for eligibility criteria, reducing recruitment timelines by up to 50% (Liu et al., 2021). This capability is particularly relevant to Finland's healthcare system, where structured and semi-structured data from Kanta Services and Apotti EHR can be leveraged to identify trial candidates efficiently.

Personalized medicine represents another frontier. AI models analyze genetic, lifestyle, and environmental data to tailor treatments to individual patients. IBM Watson for Oncology, for example, provides evidence-based treatment recommendations by cross-referencing patient data with global clinical guidelines and research (Topol, 2019). In Finland, projects such as FinRegistry are working to combine health registries on a national scale, allowing for better predictions of disease risks and more effective preventive care. This demonstrates the incredible potential of AI to improve public health strategies. However, the journey to adopting AI in healthcare isn't without its challenges. Data privacy is a major concern, especially with the EU's General Data Protection Regulation (GDPR) in effect. In Finland, the smaller population increases the risk of re-identification, which means we need strong anonymization techniques for datasets like E-lääkärinlausunto. Addi-

tionally, algorithmic bias presents ethical challenges; if models are trained on data that lacks diversity, they may overlook the needs of minority populations, worsening health disparities. A 2019 study highlighted that commercial facial recognition systems had higher error rates for darker-skinned individuals, raising concerns about equitable AI deployment (Buolamwini & Gebru, 2018).

2.4 E-lääkärinlausunto in Finnish Healthcare

E-lääkärinlausunto, or electronic medical certificates, play a vital role in Finland's advanced digital healthcare landscape. These documents serve as standardized digital records that compile crucial patient information, including diagnoses, treatment plans, medical histories, and specialist referrals, into organized or semi-organized formats. Managed by Finland's Kanta Services—a national digital health platform—E-lääkärinlausunto certificates are securely exchanged among primary care providers, hospitals, and laboratories. This ensures seamless continuity of care while strictly following data privacy regulations (Kanta Services, 2023). Unlike traditional paper records, these certificates are crafted to simplify workflows, lessen administrative tasks, and improve interoperability across Finland's decentralized healthcare network, which covers 21 hospital districts.

E-lääkärinlausunto certificates are divided into categories A, B, C, and E, each tailored for different clinical purposes. This study focuses on Type E certificates, which are thorough documents used for specialist referrals, hospital discharge summaries, and long-term care planning. They consist of unstructured free-text narratives written by doctors, detailing symptoms, diagnostic findings (such as lab results and imaging reports), prescribed medications, and follow-up recommendations. For example, a Type E certificate for a cardiovascular patient might include results from an echocardiogram, a history of hypertension, and a referral to a cardiologist. These documents are usually stored as PDFs, TIFFs, or JPEGs, often featuring handwritten notes or scanned images, which can present unique challenges for automated processing. (Finnish Institute for Health and Welfare [THL], 2022).

The variety in documentation styles—like regional dialect differences, bilingual entries (Finnish/Swedish), and inconsistent templates—makes it tricky to extract data on a large scale. For example, a diabetes diagnosis could show up as "diabetes tyyppi 2" in Finnish or "diabetes typ 2" in Swedish, which means we need strong natural language processing (NLP) models to standardize the terminology (Virtanen et al., 2021).

Data privacy is a top priority when dealing with E-lääkärinlausunto. Finland complies with the EU General Data Protection Regulation (GDPR) and the Act on the Secondary Use of Health and Social Data (552/2019), which require anonymization and limit access to identifiable patient information. For AI applications, this calls for techniques like pseudonymization and federated learning to analyze data without breaching confidentiality (Ministry of Social Affairs and Health, 2019). Still, given Finland's small population of 5.5 million, the risk of re-identification is a concern, so we must exercise careful oversight when deploying AI tools on sensitive datasets. Challenges in leveraging E-lääkärinlausunto for AI-driven solutions include:

1. **Different formats:** When you have scanned PDFs and TIFFs filled with handwritten notes, you need advanced optical character recognition (OCR) tools that are specifically designed for Finnish medical terms.
2. **Language challenges**: Bilingual entries and regional dialects require the use of multilingual NLP models, like FinBERT for Finnish and KB-BERT for Swedish (Virtanen et al., 2021)**.**
3. **Data isolation:** The disjointed storage across hospital districts limits the ability to analyze data centrally, but efforts like FinRegistry are working to integrate health data for research purposes (FinRegistry, 2023).

Despite the obstacles, E-lääkärinlausunto presents amazing opportunities for AI innovation. By applying NLP techniques such as named entity recognition (NER) and ICD-10 mapping to organize unstructured text, researchers can automate the eligibility screening process for clinical trials, track disease trends, and customize treatments. For example, if we extract "HbA1c ≥ 7%" from a diabetes patient's certificate, it could signal their eligibility for a glycemic control study. Future developments in explainable AI (XAI) and federated learning are expected to improve transparency and scalability, ensuring that Finland stays at the forefront of ethical, data-driven healthcare.NLP Techniques for Medical Text Analysis

2.5 NLP Techniques for Medical Text Analysis

NLP Techniques for Medical Text Analysis have come a long way over the years, evolving from basic rule-based systems to sophisticated deep learning models. In the early days, rule-based systems were the go-to for processing clinical texts, relying on carefully crafted rules and specialized dictionaries to pinpoint medical entities like symptoms, diagnoses, and medications. While these

systems were quite accurate in controlled settings, they often faced challenges with the variability of clinical language and needed frequent manual updates to keep up with new terms and abbreviations. Then came statistical methods like hidden Markov models and conditional random fields (CRFs), which provided a more adaptable solution. These techniques could capture the sequential relationships in text, effectively identifying multi-word medical entities and addressing the subtleties of clinical language through tailored lexicons and annotated datasets.

NLP Techniques for Medical Text Analysis are essential to the modern landscape of AI in healthcare, allowing machines to interpret and analyze unstructured clinical text. For instance, Finland's E-lääkärinlausunto contains rich medical narratives in both Finnish and Swedish. These techniques effectively transform free-text data into structured insights that support clinical decision-making and improve patient recruitment for clinical trials.

The whole process begins with text preprocessing and tokenization, where raw clinical text—often packed with abbreviations, misspellings, unconventional terms, and multilingual entries—is standardized into a consistent format. This standardization involves breaking down sentences using language-specific rules to manage the nuances of Finnish compound sentences, followed by tokenization that separates the text into words and punctuation. Because Finnish is agglutinative, we need specialized tokenizers to accurately break down complex words. Additionally, we refine the text by removing stopwords and lemmatizing, which helps filter out non-informative words and reduce inflected terms to their base forms. For instance, a sentence that describes a patient's condition is boiled down to its essential medical terms, ensuring that crucial information like symptoms, lab values, and diagnoses are kept for further analysis.

Once we get the text ready through preprocessing, Named Entity Recognition (NER) plays a key role in identifying important clinical entities like diagnoses, medications, and lab values in the narrative. This is accomplished using a combination of rule-based systems that utilize regular expressions to detect patterns such as "HbA1c > 7%," along with cutting-edge machine learning models. For example, specialized models like Fin-BERT—trained on Finnish corpora and fine-tuned for medical texts—can accurately recognize clinical entities, even when they encounter challenges like compound words that require sub-word tokenization. Techniques such as BiLSTM-CRF enhance entity detection by capturing contextual information through bidirectional LSTM networks paired with a Conditional Random Field layer, making it possible to reliably tag complex medical terms.

Temporal analysis and event extraction are essential components in clinical texts, as they reveal when diagnoses were made and how long symptoms lasted—key details for determining trial eligibility. Tools like HeidelTime, which have been customized for the Finnish language, help to normalize dates and durations, allowing for the extraction of time-specific information from clinical narratives. For example, events such as myocardial infarction or high blood pressure are identified and linked to particular dates or durations, providing a more thorough understanding of a patient's medical history.

Mapping free-text diagnoses to standardized codes is a crucial part of medical text analysis. In Finland's healthcare system, methods like rule-based dictionaries are used to manually connect everyday diagnoses to ICD-10 codes. On the other hand, more sophisticated machine learning techniques, such as transformer models specifically trained on Finnish clinical notes, can automate this mapping process. Additionally, integrating with external systems like UMLS enhances interoperability by linking medical terms to a wider standardized vocabulary, making sure that clinical data is both precise and compatible with global health standards.

The bilingual nature of Finland's healthcare system creates additional challenges, which are effectively managed through multilingual and cross-lingual NLP techniques. For example, multilingual models like mBERT can process both Finnish and Swedish text at the same time, while translation-based methods allow for a unified analysis by converting Swedish entries into Finnish. This cross-lingual proficiency ensures that all patient records are analyzed accurately, no matter the original language of the documentation.

Contextual embeddings and transfer learning take medical text analysis to the next level by capturing the subtle meanings behind clinical jargon. Pretrained language models like FinBERT and ClinicalBERT are fine-tuned on extensive datasets of clinical text, which helps them grasp the nuanced differences between similar medical terms and concepts. These models use masked language modeling to enable advanced querying and similarity matching within clinical records, making it easier to create accurate patient profiles and recruit participants for trials.

To evaluate these NLP techniques, we rely on standard metrics like precision, recall, and F1-scores for tasks such as Named Entity Recognition (NER) and text classification. We also use metrics like Cohen's Kappa to measure how well annotators agree on gold-standard datasets. For text generation tasks, we turn to metrics like BLEU and ROUGE to assess the quality of automated summaries.

Despite having these solid evaluation frameworks in place, we still face challenges like data diversity, model interpretability, and the high computational costs of deploying complex models in clinical settings. Ethical issues, including bias mitigation and GDPR compliance, are also critical. Techniques such as differential privacy and federated learning are increasingly being integrated to ensure patient confidentiality while maximizing the benefits of AI-driven analysis.

With the rise of deep learning, medical text analysis has undergone a remarkable transformation. Neural network architectures, like convolutional neural networks (CNNs) and recurrent neural networks (RNNs)—including Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs)—have shown incredible prowess in automatically learning hierarchical features straight from raw text. These models take away the burden of extensive feature engineering, enabling a more effective extraction of complex linguistic patterns. The emergence of transformer models, such as BERT and its specialized versions like BioBERT and ClinicalBERT, has further expanded the horizons of what's possible in clinical natural language processing. These pre-trained language models, fine-tuned on biomedical literature and clinical notes, shine in tasks like named entity recognition, relation extraction, and text classification by utilizing their understanding of bidirectional context. Their skill in processing text segments before and after a given point is particularly helpful in clarifying medical terms that might have different meanings based on the context.

In addition to the advancements in model architecture, there's been a lot of exciting progress in adapting NLP techniques specifically for the medical sector. Tailored tokenization and preprocessing strategies are key to addressing the unique challenges posed by clinical texts, including the use of specialized abbreviations, jargon, and the risk of misspellings. Techniques that leverage external medical ontologies and knowledge graphs are particularly helpful in resolving ambiguities and differentiating between terms that could be interpreted in multiple ways depending on the context. Another important focus is on developing privacy-preserving methods in NLP, which ensure that patient confidentiality is upheld throughout the analysis. Approaches like federated learning and differential privacy allow for training models on sensitive healthcare data without violating regulatory standards, thus encouraging innovation while respecting legal and ethical guidelines.

Integrating advanced NLP techniques into clinical text analysis really enhances our ability to extract meaningful information from unstructured data. This is particularly useful for automating the

recruitment of patients for clinical trials, as it allows for quick and accurate identification of eligibility criteria from electronic physician statements. Despite the impressive advancements, there are still challenges to face, such as data variability, making models easier to interpret, and ensuring they fit seamlessly into clinical workflows. Researchers are actively working on these challenges by investigating hybrid models that combine the strengths of rule-based systems with the flexibility of deep learning, as well as utilizing unsupervised and semi-supervised approaches to make the most of limited annotated data.

- **Neural Network Architecture:** Deep learning has truly transformed how we process clinical texts. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), along with their variations like Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), are now being used to automatically learn hierarchical features from raw text. These models shine in areas like text classification, sentiment analysis, and extracting medical entities by effectively capturing complex linguistic patterns without the need for manual feature engineering.
- **Transformer Models and Pre-trained Language Models:** The introduction of transformer architectures, such as BERT (Bidirectional Encoder Representations from Transformers), has taken clinical NLP to new heights. Models like BioBERT, ClinicalBERT, and other specialized transformers are pre-trained on extensive collections of biomedical literature and clinical notes, allowing them to grasp nuanced medical contexts and terminology. These models can be fine-tuned for specific tasks such as:
    - Named Entity Recognition (NER): Identifying clinical entities like diseases, medications, or anatomical parts with improved accuracy.
    - **Relation Extraction:** Determining relationships between entities (e.g., drug–disease interactions or symptom–diagnosis correlations).
    - **Text Classification:** Automatically classifying clinical narratives according to disease codes or severity levels.

Their bidirectional nature allows these models to capture context from both preceding and succeeding words, a crucial feature for disambiguating medical terms that might have different meanings in different contexts.

Sequence-to-Sequence Models and Summarization: Sequence-to-sequence (Seq2Seq) models, often powered by attention mechanisms, are increasingly used in tasks such as summarizing

lengthy clinical notes into concise, actionable insights. These models can help in generating summaries of patient histories, facilitating easier review by clinicians and supporting automated patient eligibility screening for clinical trials.

Embedding Techniques: Word embeddings, such as Word2Vec and GloVe, have long been employed to map words to dense vectors that capture semantic relationships. In the medical domain, embeddings are often refined or entirely re-trained on specialized datasets (e.g., MIMIC-III) to better represent the vocabulary and semantics unique to clinical language. Contextual embeddings derived from models like ELMo and transformer-based embeddings have largely supplanted static embeddings due to their ability to generate context-aware representations.

2.6 AI-Driven Recruitment Systems

AI-driven recruitment systems are transforming the landscape of clinical trial enrollment by automating the identification of eligible patients and streamlining the recruitment process. These systems leverage advanced machine learning algorithms, natural language processing (NLP) techniques, and sophisticated data analytics to extract meaningful information from large volumes of unstructured clinical data, such as electronic health records and medical certificates (Chopra et al., 2023). By integrating Optical Character Recognition (OCR), Named Entity Recognition (NER), ICD mapping, and temporal analysis, these platforms effectively convert free-text data into structured, actionable insights. This automated approach not only minimizes the time and costs involved in manual screening but also increases the accuracy of patient matching, ensuring that only individuals who meet the specific clinical criteria are considered for further review.

For example, systematic reviews have indicated that AI-based screening can reach sensitivity rates of up to 90.5% and specificity rates close to 99.3% (Chow et al., 2023). This level of performance is crucial for speeding up the recruitment phase, particularly in trials where every second counts. Moreover, AI-driven systems have the ability to learn and adapt continuously based on feedback, enhancing their predictive skills over time. This ongoing improvement is especially beneficial in varied clinical environments, where patient demographics and disease presentations can differ significantly.

Another impressive case is the use of Viz RECRUIT, an AI recruitment platform utilized in the EMBOLISE trial. This system not only analyzed thousands of imaging studies to identify subdural

hematomas but also facilitated communication among the research team, resulting in a 36% boost in enrollment rates (Hassan et al., 2023). These platforms highlight the real-world advantages of AI in minimizing screen failures and ensuring that recruitment is both timely and efficient.

3 Research Methdology

The methodology outlined in this study is aimed at developing, implementing, and evaluating an AI-driven system that automates patient recruitment by applying NLP techniques to unstructured Finnish medical documents. This approach is closely aligned with the Design Science Research (DSR) paradigm, which is particularly suited for research that seeks to create and assess IT solutions designed to address identified practical problems.

3.1 Methodology

This study utilizes the Design-Science Research Methodology (DSRM) as defined by Hevner et al. (2004). According to DSRM, research is about intentionally creating and carefully evaluating innovative artifacts that address relevant, real-world problems, which in turn enhances the capabilities of individuals and organizations.

Design-Science Research Methodology (DSRM) isn't categorized as "qualitative" or "quantitative." Rather, it's a framework aimed at creating an artefact (the "design" element) and then evaluating that artefact with the most appropriate empirical techniques. Hevner et al. explain that "A mathematical basis for design allows many types of **quantitative evaluations** … The further evaluation of a new artefact … affords the opportunity to apply empirical and **qualitative** methods." They also list various evaluation options that range from observational case studies (qualitative) to controlled experiments and optimization proofs (quantitative).

According to Peffers et al. (2007), the Design Science Research Methodology (DSRM) is all about following a structured and iterative process. This includes steps like identifying the problem, setting objectives for the solution, designing and developing the artifact, demonstrating how it works, evaluating its performance, and sharing the results. This approach fits perfectly with the focus of this study, which introduces an innovative NLP-based pipeline aimed at automating the extraction of eligibility criteria from E-lääkärinlausunto documents.

3.2 Research approach

Design-science research (DSR) sees knowledge creation as a continuous cycle of building and evaluating purposeful IT artifacts. Hevner and his colleagues lay this out through seven guidelines that cover aspects like problem relevance, rigor, and a thorough evaluation of the artifact's functionality, performance, and usability. Their list of evaluation strategies—such as case studies, simulations, controlled experiments, optimization, and informed arguments—serves as the analytical framework that our study will later utilize when we assess precision, recall, and workflow impact.

The recent study by Abbasi et al. (2024) highlights the importance of Design Science Research Methodology (DSRM) in developing AI systems. They introduce a framework that helps design researchers navigate the complexities of creating AI artifacts, tackling issues like rapid advancements, evaluation uncertainties, and the integration of sociotechnical factors. Their pathways framework promotes research that not only focuses on building effective AI systems but also enhances our understanding of how these systems interact with users, various domains, and their environments. This study follows that pathway by examining both the technical feasibility and clinical applicability of AI in the context of Finnish healthcare.

Rahmanian et al. (2024) showcase a fascinating example of how AI techniques—particularly prompt-based learning with large language models—can be utilized for selecting cohorts in clinical trial recruitment. Their research highlights the power of ontology-based summarization and GPT-3.5 in sifting through eligibility criteria found in narrative medical records. In a similar vein, this study introduces a BERT-based and CRF-based NLP pipeline tailored for the Finnish language and healthcare documents, effectively bridging the divide between technological advancements and practical clinical applications through a well-structured design science approach.

Traditional statistical techniques, such as K-fold cross-validation and paired t-tests, often don't provide reliable comparisons when evaluating NLP models, particularly in cases with skewed data or complex metrics like F1 (Wang & Li, 2019). To enhance the robustness of model comparisons in this study, we explored the Bayesian testing approach suggested by Wang and Li (2019). Their method features a block-regularized 3×2 cross-validation procedure and estimates posterior distributions for evaluation metrics, leading to more trustworthy conclusions about which model (like

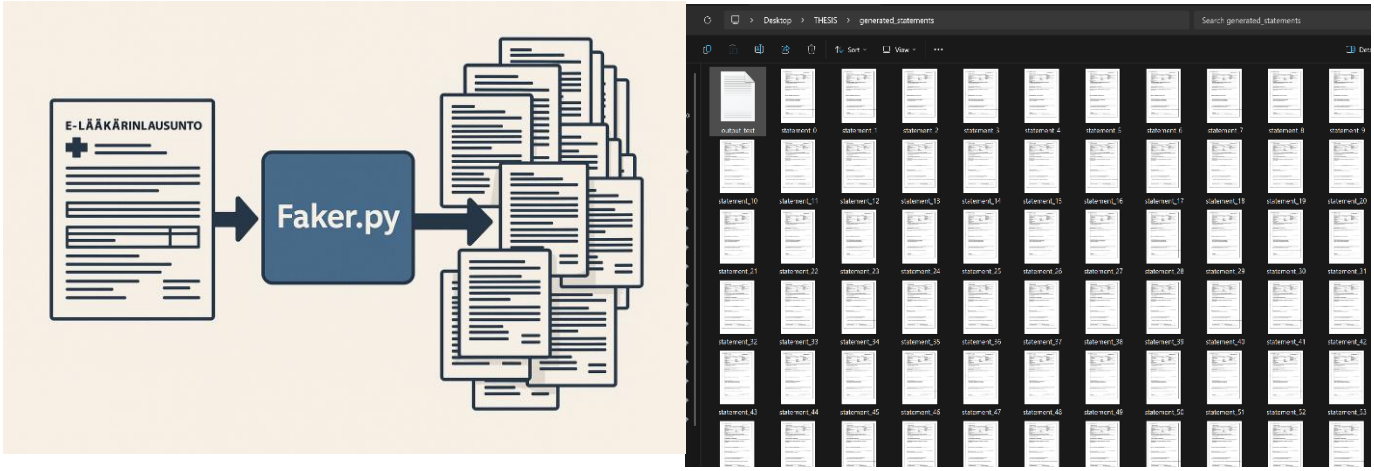
CRF vs. BERT) truly outperforms the other. While we didn't implement it in its entirety, this statistical insight guided our evaluation design and underscores the necessity for future work in adopting Bayesian validation for clinical NLP applications.

3.3 Data Sources - Generating Synthetic Data

The primary data source comprises synthetic E-lääkärinlausunto documents generated via a Python script. This script uses a template image file for the certificate layout, a CSV file containing valid ICD-10 codes, and randomized patient attributes generated through the Faker library. Such synthetic data mirrors real medical documentation in structure and content, including bilingual or Finnish-specific entries and variable formatting. Though synthetic, these documents serve as a valuable test bed for AI models to handle noise, typographical errors, and domain-specific jargon. In future stages, select real (and appropriately anonymized) E-lääkärinlausunto samples may be incorporated to further validate model performance and generalizability (Chopra et al., 2023; Klassen, 2016).

Figure 3 demonstrates the synthetic data generation process using a custom faker.py script. Starting from a sample E-lääkärinlausunto template, the script programmatically generates hundreds of anonymized clinical statements that mimic real-world structure, enabling safe and scalable training for the downstream NLP pipeline without exposing sensitive patient information.

Figure 3: Visual presentation of synthetic data creation

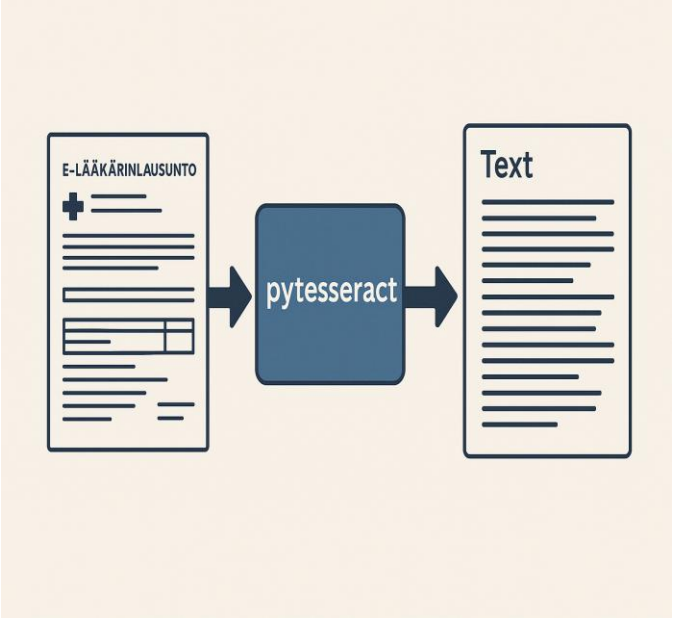


3.4 Optical Character Recognition (OCR) With Pytesseract

OCR is the foundational step in digitizing any image-based document, such as scanned E-lääkärinlausunto forms. Tools like Tesseract or pdfplumber are employed to detect and transcribe text from potentially noisy or skewed images (Hassan et al., 2023). Preprocessing workflows—such as image deskewing, denoising, and binarization—enhance accuracy, which is vital when dealing with handwritten annotations or low-resolution scans. Additionally, language-specific models for Finnish characters, diacritics, and compound words further refine output. This OCR phase underpins all subsequent NLP tasks, ensuring that text is readily machine-readable (Chow et al., 2023).

Figure 4 illustrates the OCR extraction workflow, where scanned E-lääkärinlausunto documents are converted into machine-readable text using the pytesseract engine. This step is crucial for enabling downstream NLP processing on unstructured clinical content originally available only in image format.

Figure 4: Output from OCR



ITA Date: 2023-(PP84 00090 SKANNATTU

MAKSUSITOUMUSPÄYNTÖ E- LÄÄKÄRINLAUSUNTO – sivut
Työtapaturma [] Ammattitauti [] Tutkimus ammattitautiepäilyn vuoksi | Yksityistapaturma
Liikennevahinko |] Potitasvatineo [ ] Sotilastapaturma. X) Muu, mikä?

Vakuutusyhtiö Vahinkonumero Vakuutusnumero

Pohjola Vakuutus Oy £ 40-999-999-0

"Sukunimi ja etunimet Henkilötunnus

Patient Name: Santeri Oksanen Patient ID: 050304-5726

Lähiosoite Postinumero Positolmipaikka

Kalevantie 250 48600 KOTKA

Ammatti Työnantala

Asentaja Mesikäpälä Oy

4.| Tapaturman tai sairauden alku:
Pym, klo ja paikka

Tapaturman kuvaus tai sairauden alkaminen. Altistumistiedot (tarvittaessa kohdassa 7 tai erillisellä liitteellä)

43-v mies, joka harrastaa kestävyysjuoksua. Sääriluun rasitusmurtuma todettu 9/2019 oireilee edelleen vuoden vaihteessa ja mri kuvalla halutaan varmistaa luun paranemisen tilanne 5

3.5 Manual Annotation using Label Studio

With the text data prepared (now in plain text form after OCR and cleaning), the next step was to **annotate the data for named entity recognition**. We needed to label the entities of interest that

our models should learn to identify. Based on the trial recruitment use-case, we decided on the following entity categories:

- **Diagnosis/Condition**: medical conditions, diseases, or diagnoses that the patient has (e.g., *tyypin 2 diabetes*, *astma*, *kaksisuuntainen mielialahäiriö* for bipolar disorder, etc.). This category also included any explicit mention of an ICD-10 code, tagging it as part of the diagnosis entity.

- **Medication**: any medications or active treatments mentioned (drug names like *metformiini*, *insuliini*, *bisoprololi*, or therapy like *kemoterapia* for chemotherapy).

- **Symptom/Finding**: any symptom or finding noted (e.g., *korkea verensokeri* for high blood sugar, *väsymys* for fatigue). We included this category so that models can identify symptomatic descriptors that might relate to eligibility (some trials might require a symptom severity).

- **Procedure/Test**: if any mention of a medical procedure or test result (for example, "HbA1c 8.5%" or "MRI imaging done"), though our synthetic data had few of these, we allowed a tag for it.

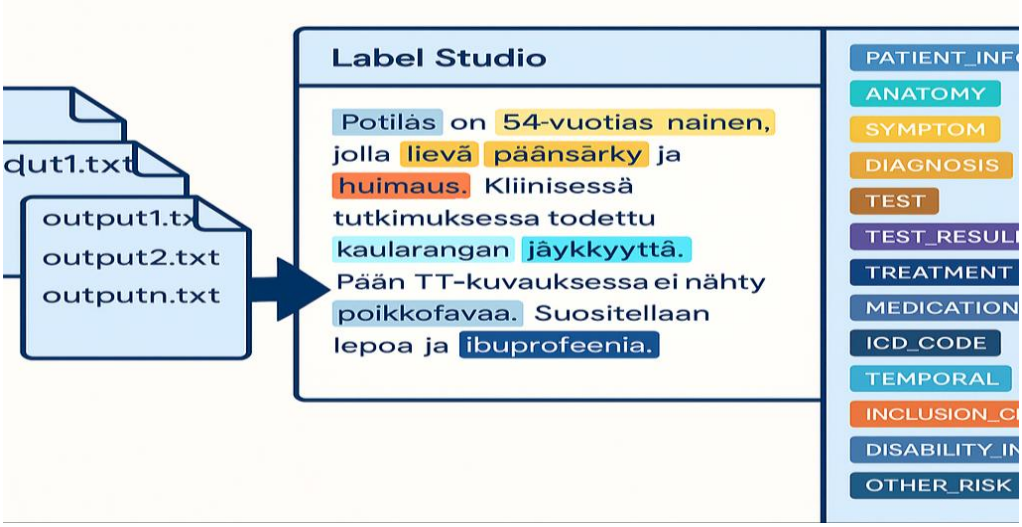
- **Other**: We had an "Other" tag for any miscellaneous but potentially relevant info (like lifestyle factors if present, e.g., *"polttanut 20 vuotta"* – "has smoked for 20 years" – which could be relevant to some trials). In practice, this was rarely used.

We used the **BIO (Begin, Inside, Outside)** tagging scheme for the entities, as is standard for NER tasks. That means if an entity spans multiple tokens, the first token gets a B-<EntityType> tag and subsequent tokens get I-<EntityType>. For example, for "iskiashermon tulehdus" (sciatic nerve inflammation), which is a two-word diagnosis, "iskiashermon" would be tagged B-Diagnosis, and "tulehdus" I-Diagnosis. Single-token entities are just labeled as B-<type> (with no I following).

To perform the annotation, we set up a project in **Label Studio**, an open-source data labeling tool. We loaded each cleaned text (from the synthetic documents) as a separate task in Label Studio.

We defined the label set according to the categories above. Two annotators (including the author) went through each text and highlighted spans corresponding to the entities, assigning the appropriate label. We also labeled the ICD codes when present; for instance, in “Diagnoosi: E11 Tyypin 2 diabetes”, the token “E11” and “Tyypin 2 diabetes” together were marked as a single Diagnosis entity (Label Studio allows labeling a span covering multiple discontinuous parts if needed, but in this case we usually labeled the contiguous text “E11 Tyypin 2 diabetes” as one span). We aimed for high recall in annotation – meaning we tried to label all medically relevant terms, even if they might not be directly needed for a particular trial. This enriches the training data and lets the model learn a broad range of entities. The labeling guidelines (see Appendix B) provided examples to ensure consistency, particularly around edge cases like overlapping entities or nested info. For instance, if a sentence said “Potilaalla on diagnosoitu nivelreuma ja siihen liittyvä keuhkofibroosi” (“The patient has been diagnosed with rheumatoid arthritis and related pulmonary fibrosis”), we label “nivelreuma” as Diagnosis and “keuhkofibroosi” also as Diagnosis (even though the latter is a complication of the former). We did not attempt to encode the relationship (that fibroosi is secondary); we only marked the entity spans themselves. Figure 5 we can see visually manual annotation of patients data using label studio.

Figure 5: Manual annotation



After completing annotations for all documents, we exported the labeled data in a **CoNLL format** using Label Studio’s export function. The initial export had each token on a line with its label, along with some additional columns (like part-of-speech tags) automatically generated by Label Studio’s

internal NLP features. For example, Label Studio might output lines like 1 Potilas NN O (where 1 is token index, Potilas the token, NN the POS, and O the NER tag). We ended up with a file where each sentence is separated by a blank line, following the CoNLL convention, and each token line contains multiple columns.

3.6 Converting Annotations to CoNLL Format

The raw export from Label Studio needed some processing to be in the exact format expected by our training scripts. In fact, we needed two slightly different formats: one for the CRF training and one for the BERT training.

For the CRF model: We utilized a popular Python CRF implementation (sklearn-crfsuite). We wrote a parser to read a CoNLL-style file and create sequences of tokens with features. Our CRF training code expected each line to have at least a token and a label, but we also wanted to leverage part-of-speech (POS) and other features if available. The Label Studio export had 5 columns (token index, token text, POS, chunk tag, NER tag) similar to CoNLL-2003 format. We decided to pad this to a fixed number of columns (9 columns as a standard, where unused ones are just underscores) for safetyfile-kd3psdybkmwlz6kwf5d2y3file-kd3psdybkmwlz6kwf5d2y3. Essentially, we created a "fixed" conll file where each line is: TOKEN POS CHUNK NER _ _ _ _ _. The NER column (the fourth in this case) holds the BIO tag for the entity (or "O" for outside). This padding was done to ensure consistent column indexing in our CRF feature extraction code.

We developed a small script fix_conll_crf.py (Appendix A.1) for this. It reads the Label Studio export line by line:

- If a line is empty or a document separator, it writes an empty line to output (to preserve sentence boundaries).

- If a line begins with "-DOCSTART-" (a convention sometimes present in CoNLL exports), it leaves it as is (or pads it appropriately).

- For regular token lines, it splits by whitespace. If the line has fewer than 9 columns, it pads with "_" (underscore placeholders) until 9 columnsfile-kd3psdybkmwlz6kwf5d2y3. If it has more, it trims it (though in our case it never had more than 5).

- Then it writes the line out.

The result was a file (let's call it annotated_data_crf.fixed.conll) ready for CRF consumption.

For the BERT model: We needed a simpler two-column format: TOKEN <space> TAG. The Hugging Face Transformers library can work with datasets prepared as lists of tokens and labels in Python, but for simplicity we decided to also create a textual CoNLL where each line has a token and its BIO tag, and sentences separated by blank lines. We wrote another converter fix_conll_bert.py (Appendix A.2) which:

- Reads each line, ignores any line starting with -DOCSTART- or empty lines indicating sentence breaks (but writes a blank line to output to mark the sentence boundary)file-3aggpldx9gppnq5mzsutbh.

- For each token line, it splits on tab (or whitespace) to get columnsfile-3aggpldx9gppnq5mzsutbh.

- If the line had the 5-column LS format, it takes column 2 as the token and the last column as the NER tagfile-3aggpldx9gppnq5mzsutbh. If the export were already two-column (token and tag), it can handle that too.

- Writes out "TOKEN TAG" to the output filefile-3aggpldx9gppnq5mzsutbh.

This gave us a file annotated_data_bert.conll containing lines like:

```
Potilas O
on O
diagnoitu O
tyypin B-Diagnosis
2 I-Diagnosis
diabetes I-Diagnosis
ja O
```

3.7 Training NER Models (CRF and BERT)

Now we proceed to train the two models using the prepared data. The goal was to train on a portion of the synthetic data and evaluate on a held-out portion, to compare model performance. We roughly did an 80/20 split: 80% of the documents for training, 20% for testing. Given our dataset size (~50 documents), the test set might be small, but sufficient for a proof-of-concept evaluation.

3.7.1 CRF Model Training

For the CRF, we implemented a trainer in Python (CRF_Trainer.py – see Appendix A.3 for snippet). The CRF is a sequence model that relies on manually crafted features for each token. Based on literature and some trial-and-error, we included the following features for each token:

- Lowercased token text (to handle capitalization differences, e.g., “Astma” vs “astma”).
- Lemma of the token (we utilized the Finnish spaCy model fi_core_news_sm or lg to get lemmas and part-of-speech)file-u6epzl7r3d7mismxg5bdnafile-deu3rbbxshrkyt3fju6exd. Finnish words vary a lot by case and suffix; lemmas help normalize that (e.g., “diabetesta” -> lemma “diabetes”).
- Part-of-speech tag of the token (noun, verb, etc.)file-u6epzl7r3d7mismxg5bdna.
- Suffix of length 3 (last three characters)file-u6epzl7r3d7mismxg5bdna – this can sometimes help, e.g., many Finnish condition names end in “-itis” in Latin or “-tauti” in Finnish, etc.
- Shape feature: a simplistic regex-based shape (e.g., “Xxx” or “dd.mm.yyyy” for dates, etc.) file-u6epzl7r3d7mismxg5bdna.
- Boolean features indicating if the token is at the beginning of a sentence (BOS) or end of a sentence (EOS)file-u6epzl7r3d7mismxg5bdna. Sometimes entity distribution can depend on position.
- Previous token’s lemma (and the one before that)file-u6epzl7r3d7mismxg5bdna, and next token’s lemma (and the one after that)file-u6epzl7r3d7mismxg5bdna – providing context window of 2 on each side.
- Morphological features: spaCy provides rich morphology for Finnish (e.g., case, tense, person, etc.). We included all morphological attributes for the token as separate featuresfile-

u6epzl7r3d7mismxg5bdna. For example, a token might have features like morph_Case=Partitive or morph_Number=Sing.

- **ICD dictionary match**: We loaded a dictionary of ICD-10 codes with their Finnish descriptions (from a CSV provided by THL, the Finnish Institute for Health), mapping codes to disease names. Our function icd_lookup(tok_text, icd_dict) returns True if the token (after stripping punctuation) is an ICD code present in the dictionaryfile-deu3rbbxshrkyt3fju6exd file-deu3rbbxshrkyt3fju6exd. We added a feature “icd_matched”: True if sofile-deu3rbbxshrkyt3fju6exd. The idea is that if a token exactly matches an ICD code like “E11” or “C50”, that is a strong signal it’s part of a Diagnosis entity. The CRF can learn this association.
- We then used sklearn_crfsuite.CRF() to train a modelfile-u6epzl7r3d7mismxg5bdna. We set it to use L-BFGS optimization, with L1 and L2 regularization (hyperparameters c1=0.1, c2=0.05 chosen by quick experimentation). The training data for CRF was prepared by reading the .fixed.conll file: splitting into sentences and corresponding label sequences. We split those into train and test (80/20 as mentioned)

```
{
 "token.lower": "diabetes",
 "lemma": "diabetes",
 "pos": "NOUN",
 "suffix3": "tes",
 "shape": "aaaaaaa" (just letters),
 "BOS": False,
 "EOS": False,
 "prev_lemma": "2" (from "tyypin 2"),
 "prev2_lemma": "tyyppi",
```

Figure 6 depicts the CRF model training workflow, where annotated clinical data in CoNLL format is processed through a custom training script crf_trainer.py to produce a serialized model file crf_model.pkl used for clinical entity recognition.

Figure 6: CRF training

3.7.2 BERT Model Training

- For the BERT-based model, we used the Hugging Face Transformers library. We decided to start with the multilingual BERT (bert-base-multilingual-cased) since it supports Finnish out-of-the-box, and we also wanted to experiment with a Finnish-specific model (FinBERT). Initially, we fine-tuned the multilingual BERT on our data.

  We wrote a script bert_trainer.py (Appendix A.4) to handle this training. The steps involved:

- Reading the annotated_data_bert.conll file into a list of {"tokens": [...], "ner_tags": [...]} for each sentencefile-fxgn696dyexpvqt2awjnwy. We converted the BIO labels into numeric IDs (since the model expects numeric class labels). We created a label list like ["B-Diagnosis", "I-Diagnosis", "B-Medication", ... "O"] and a mapping label2id and id2labelfile-fxgn696dyexpvqt2awjnwy.
- We then loaded a pre-trained tokenizer and model. For tokenizer and model initialization, one approach is to use a model checkpoint name. We started with:

Figure 7: Bert training

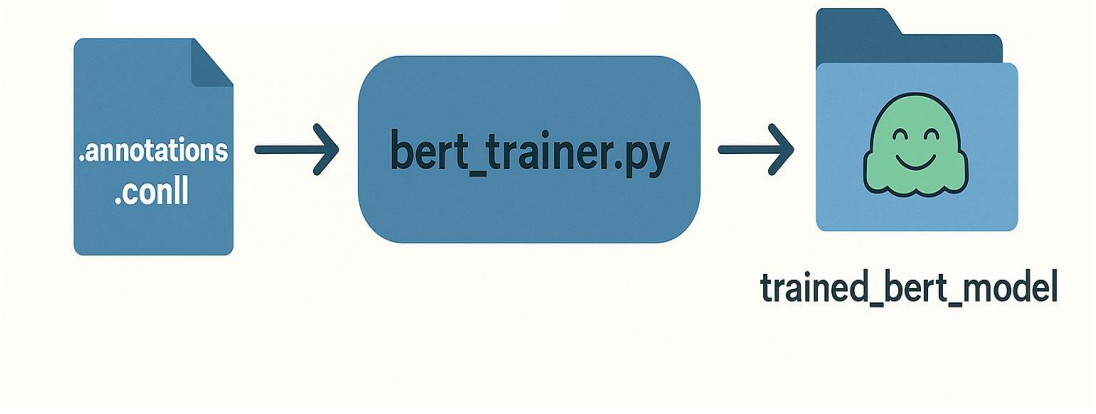


Figure 7 shows the BERT model training workflow, where annotated clinical text in CoNLL format is passed into a fine-tuning script bert_trainer.py to produce a domain-adapted transformer model stored as trained_bert_model for clinical entity recognition.

3.7.3 Named Entity Recognition (NER)

NER modules identify clinically relevant elements (diagnoses, medications, patient demographics, etc.) within extracted text. The approach can vary from rule-based systems—using specialized dictionaries for medical terms—to advanced deep learning models like transformer-based architectures (BERT, BioBERT, or FinBERT). Given Finnish's agglutinative morphology and potential bilingual entries, domain adaptation is crucial. For instance, a phrase like "diabetes tyyppi 2" must be recognized and normalized to "type 2 diabetes." These labeled entities form the basis for downstream tasks such as trial eligibility filtering and temporal reasoning (Klassen, 2016).

3.7.4 ICD Mapping

After identification, key terms are mapped to standardized medical codes—particularly ICD-10—in order to unify diverse clinical expressions under a single classification framework (Chopra et al., 2023). Automated mapping can be achieved using rule-based matching, fuzzy string matching, or machine learning algorithms trained on annotated corpora. This process is critical for trial criteria that specify particular disease codes or comorbidities.

3.7.5 Temporal Analysis

Temporal analysis extracts information on event timing—diagnosis dates, symptom onset, or treatment durations—from the text. Methods such as HeidelTime or transformer-based models adapted for Finnish can normalize time expressions to structured formats (Chow et al., 2023). This is pivotal in clinical trials where patient eligibility may hinge on recency of diagnosis or length of symptom history. Integrating temporal data allows researchers to build patient timelines, enhancing the accuracy of AI-based screening for ongoing or upcoming studies.

3.8 Overall about training CRF and BERT Model

- Data preparation: The tricky part with BERT is that it operates on sub-word tokens. For example, a word like “sydämen” might be broken into two WordPiece tokens by the BERT tokenizer. We need to align our labels to these sub-tokens. We wrote a function tokenise_align that takes the list of words in a sentence and the list of labels, and uses the tokenizer to split the words while keeping track of alignmentfile-fxgn696dyexpvqt2awjnwy. The strategy was: for each original word, assign its label to the first sub-token and assign a special label ID (-100) to subsequent sub-tokens, so that these will be ignored in the loss calculation (this is a common practice in token classification with transformers). In code, we achieved this by using the word_ids() function after tokenization to map sub-token indices back to word indicesfile-fxgn696dyexpvqt2awjnwy. If word_ids() returns None (for special tokens like [CLS], [SEP]) we assign -100. If it returns an index, we assign the corresponding label ID for that word index, except if it’s the same as the previous sub-token’s word index (meaning this sub-token is a continuation of the previous word) – in that case we also assign -100file-fxgn696dyexpvqt2awjnwy. This effectively tells the model to only learn from the first sub-token of each word.

- We created a HuggingFace Dataset and applied this mapping function to get tokenized inputs with aligned labelsfile-fxgn696dyexpvqt2awjnwy. We then set the dataset format to PyTorch.

- Training configuration: We used the Trainer API from HuggingFace. We set the training arguments such as number of epochs (we chose 20, given the small dataset, to ensure convergence)file-fxgn696dyexpvqt2awjnwy, batch size (8), learning rate (5e-5, a common fine-tuning rate), and disabled saving checkpoints at each epoch (since the dataset is small) to streamline itfile-fxgn696dyexpvqt2awjnwy.

- We then invoked Trainer with our model, dataset, and a data collator that pads sequences dynamically (Transformers provides DataCollatorForTokenClassification to pad inputs to the same length in a batch)file-fxgn696dyexpvqt2awjnwy. We trained the model using .train()file-fxgn696dyexpvqt2awjnwy. The training process took only a few minutes on our hardware given the small data size and model.

- After training, we saved the fine-tuned model and tokenizer to disk (./bert-ner-output directory)file-fxgn696dyexpvqt2awjnwy. This directory contains the model weights, which we can load later for inference in the pipeline.

- Since our CRF was evaluated on a test set, we wanted to evaluate BERT similarly. We did not explicitly write an evaluation loop in the trainer script (we set evaluation_strategy = "no" for simplicityfile-fxgn696dyexpvqt2awjnwy). Instead, after training, we manually tested the model on the hold-out texts using our pipeline code (described in the next chapter). We considered metrics such as overall token-level F1 and also looked at entity-level precision/recall by comparing model outputs to ground truth on the test set. (In hindsight, integrating the evaluation in the Trainer or using Trainer.evaluate() with a small eval dataset would have been more straightforward, but given the small data, the manual approach was sufficient.)

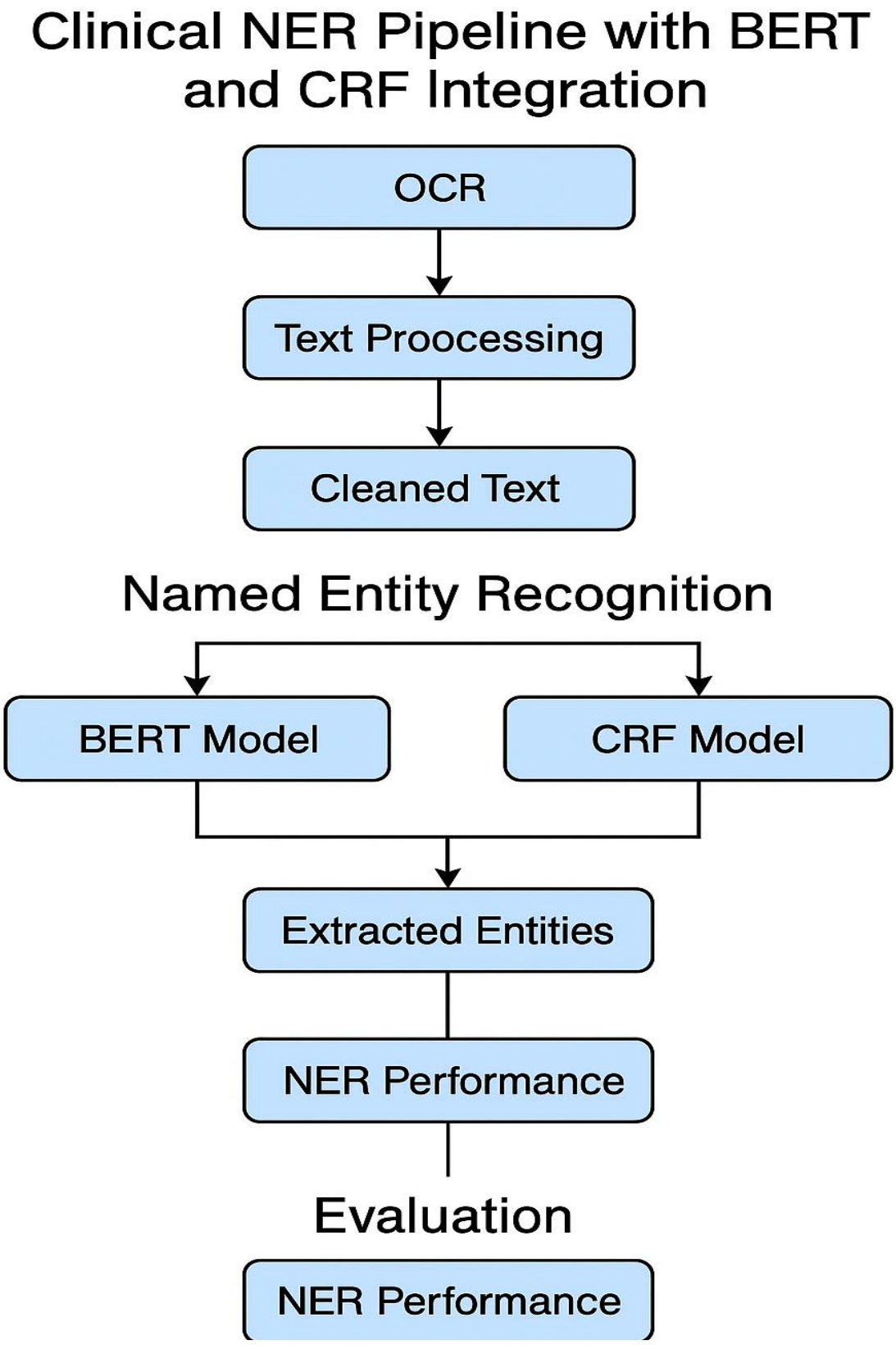
One consideration: We fine-tuned using the multilingual BERT which already has some Finnish capability, but not specialized in medical text. In future work (or as an extension), we could initialize from **FinBERT** (the model by Virtanen et al.), which might capture Finnish linguistic nuances better. FinBERT was trained on a large Finnish corpus (including internet and news text). While it's not domain-specific to medicine, its vocabulary might align slightly better with Finnish terms. We did attempt a quick experiment switching the from_pretrained to a FinBERT checkpoint (available via HuggingFace model hub as "TurkuNLP/bert-base-finnish-cased-v1" for instance). The training procedure remains identical. We report on any differences observed in Chapter 5.

Finally, to ensure a fair comparison, we used the same training and test split for both CRF and BERT. Both models were effectively trained on the same data distribution. The CRF had the advantage of using some external knowledge (ICD dictionary), whereas BERT had the advantage of a powerful pretrained language representation. The stage was set to see which would perform better on identifying our entities in Finnish doctors' statements.

With our models trained and saved, we then integrated them into an end-to-end pipeline. The next chapter on Architecture will detail how the pieces (OCR, CRF model, BERT model) come together to form a complete system, along with a visual overview.

4 Architecture

Figure 8: Architecture design



4.1 System Flowchart

The pipeline begins with the **Input Document**, typically a scanned image or PDF of a doctor's statement. The first module is **OCR (Optical Character Recognition)**, which converts the image into raw text. Once text is obtained, it is cleaned and normalized. This clean text is then passed in parallel to two subsystems:

- The **CRF-based NER subsystem**, which includes tokenization, feature generation (using spaCy and an ICD-10 dictionary), and the CRF model that labels each token with an entity tag.

- The **BERT-based NER subsystem**, which includes a BERT tokenizer that splits text into sub-word tokens, and the fine-tuned BERT NER model that labels tokens (with an aggregation step to combine sub-word outputs into full token labels).

Both subsystems output a list of identified entities with their categories. For example, the CRF might output: {"Tyypin 2 diabetes": Diagnosis, "metformiini": Medication}, and the BERT might output a similar set (ideally). These results can be post-processed or directly sent to a **Matching Engine** (outside the scope of this study, but conceptually, that engine would compare these entities against a given trial's inclusion/exclusion criteria to flag a match).

The architecture is modular. Each part (OCR, CRF pipeline, BERT pipeline) can be maintained or upgraded independently:

- If a higher-accuracy OCR becomes available, it could replace Tesseract.
- If more training data is obtained, the CRF and BERT models can be retrained without changing the rest of the system.
- The dual pipeline approach also allows ensemble usage – one could combine the outputs (e.g., take union of entities) to potentially increase recall.

Next, we detail the CRF and BERT pipelines as implemented.

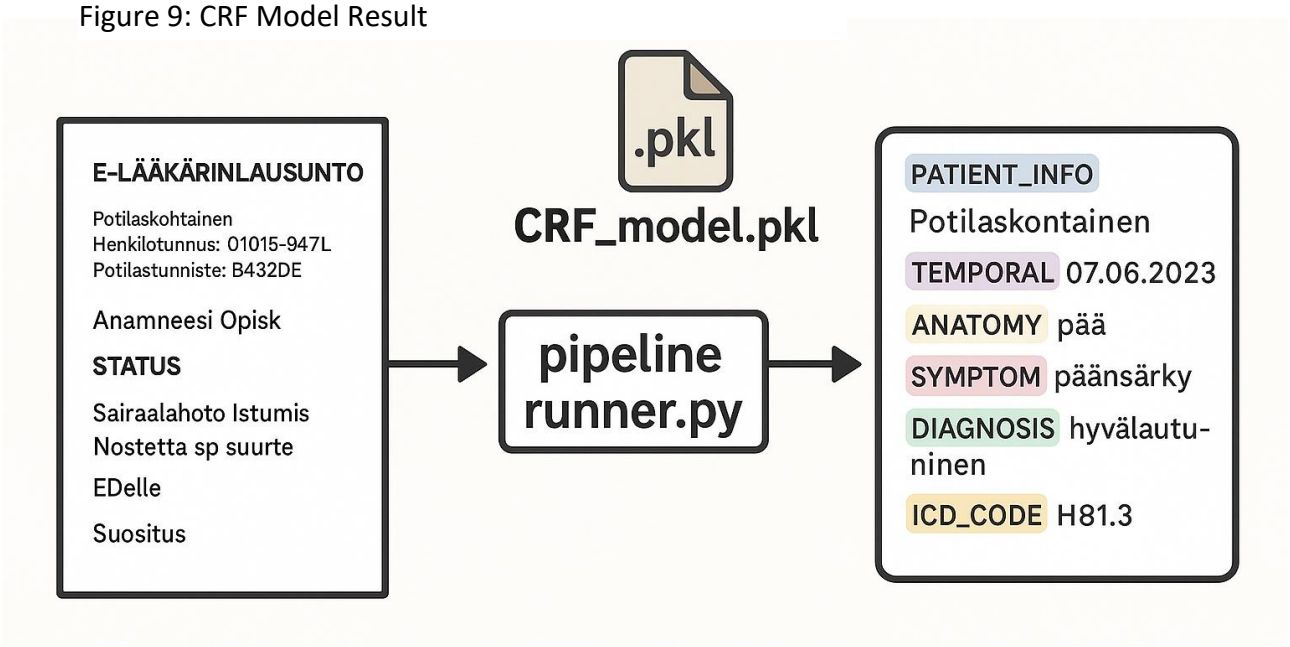
4.2 CRF Pipeline Description

The CRF pipeline processes the text through a series of steps, many of which we implemented in the pipeline_runner.py script. From the figure 9 we can visualize CRF pipeline and the script extracts structured clinical entities such as symptoms, diagnoses, dates, and ICD codes from the input narrative for downstream eligibility analysis. The flow is as follows:

1. **Loading Resources**: When the pipeline starts, it loads necessary resources into memory. This includes the spaCy Finnish model (we attempt to load the large model fi_core_news_lg, falling back to fi_core_news_sm if unavailable)file-deu3rbbxshrkyt3fju6exdfile-deu3rbbxshrkyt3fju6exd. It also loads the ICD-10 dictionary from a CSV file into a Python dictionary for quick lookupfile-deu3rbbxshrkyt3fju6exd. The CRF model (which was trained and saved as a pickle file) is loaded using joblib.loadfile-deu3rbbxshrkyt3fju6exd. This loading is indicated by the boxes "Load spaCy NLP" and "Load ICD Dictionary" and "Load CRF model" in the flowchart (left side).
2. **Tokenization and Feature Extraction**: The input text (which might be a few sentences) is split into tokens. In our implementation, we actually split on whitespace for simplicity and then rely on spaCy to get linguistic annotations for each tokenfile-deu3rbbxshrkyt3fju6exd. Each token is converted into the feature representation as described in Chapter 3. This is done by the function build_crf_features(tokens, icd_dict)file-deu3rbbxshrkyt3fju6exdfile-deu3rbbxshrkyt3fju6exd. It iterates token by token and creates the feature dictionary (like lemma, pos, etc.), and checks the ICD dictionary for a matchfile-deu3rbbxshrkyt3fju6exd. The output of this step is a list of feature dicts, one per token, preserving order. This corresponds to the "Feature Engineering (spaCy + ICD)" box in the flowchart, where spaCy provides lemma/POS/morphology and the ICD list adds a flag.
3. **CRF Model Prediction**: The list of feature dicts is then fed to the CRF model's predict_single methodfile-deu3rbbxshrkyt3fju6exd. The CRF model returns a list of labels (one per token) – these are the BIO tags indicating each token's entity class or "O". The pipeline then pairs each token with its predicted labelfile-deu3rbbxshrkyt3fju6exd. For instance, it might produce something like: [("Potilas", O), ("on", O), ("diagnoitu", O), ("tyypin", B-Diagnosis), ("2", I-Diagnosis), ("diabetes", I-Diagnosis), ...].

4. **Post-processing**: The sequence of token-label pairs is then processed to group tokens into full entities and filter out non-entity tokens. Essentially, we scan through and whenever we see a B- tag, we take that token and any subsequent I- tags of the same category to form an entity phrase. In our running example, “tyypin 2 diabetes” would be grouped as one Diagnosis entity. The output is then a structured list of entities, e.g., [{"text": "tyypin 2 diabetes", "label": "Diagnosis"}, {"text": "metformiini", "label": "Medication"}, ...]. In pipeline_runner.py, for simplicity, we actually just printed each token with its labelfile-deu3rbbxshrkyt3fju6exd, but in a real application we would assemble them as described.
5. **Output**: The CRF pipeline outputs the entities and prints a summary. In our demo prints, we showed each token and the label in a columnar format for inspectionfile-deu3rbbxshrkyt3fju6exd.

Figure 9: CRF Model Result



One advantage of the CRF pipeline is that it’s relatively lightweight. The model file was only a few megabytes, loading is fast, and prediction on a document with tens of tokens is virtually instantaneous (CRF inference is linear in sequence length, with a small constant). The feature engineering step adds some overhead (calling spaCy for each token), but since documents are short, it’s not an issue. If scaling to very large texts, one could optimize by using spaCy to do a full parse of the text (so it tokenizes and gives all tokens and their attributes in one go, which we partially did by raw_tokens = text.split() then calling nlp(tok)[0] on eachfile-deu3rbbxshrkyt3fju6exd – this is not the most efficient because we call the NLP model on each token independently rather than the whole text; a refinement could process the whole text at once through spaCy).

The CRF pipeline is also interpretable to a degree – one can inspect the learned feature weights (not done here, but possible) to see what cues the model learned for each entity. For example, it might have a high weight for lemma=="diabetes" -> Diagnosis, or icd_matched==True -> Diagnosis.

4.3 BERT Pipeline Description

The BERT pipeline follows a different approach, leveraging the transformer model's contextual embeddings rather than manual features. From the figure 10 we can visualize BERT pipeline and the trained model to extract structured entities such as patient information, diagnosis, and medication using the pipeline_runner.py script. The steps in this pipeline (as implemented in pipeline_runner_bert.py, see Appendix A.5 snippet) are:

1. **Loading the BERT Model**: At startup, the pipeline loads the fine-tuned BERT model and its tokenizer from the saved directory. In code, we did:

```
tokenizer = AutoTokenizer.from_pretrained(model_path)
model = AutoModelForTokenClassification.from_pretrained(model_path)
```

where model_path points to our bert-ner-output directoryfile-fhrmrm7tm4fjwj1adsf1j8. This loads the model with the architecture (including the classification head with the correct number of labels and the trained weights). The flowchart's right side shows this as "Load BERT + Tokenizer".

2. **OCR Text Input**: The BERT pipeline uses the same OCR text as the CRF pipeline (in our implementation, we run them in one script, so they share the OCR output). We apply the same cleaning to ensure the text has uniform spacing.
3. **NER Pipeline Inference**: We utilized Hugging Face's high-level pipeline for NER:

```
nlp_ner = pipeline("ner", model=model, tokenizer=tokenizer, aggregation_strategy="simple")
predictions = nlp_ner(text)
```

This pipeline does a lot under the hood: it will take the input text, tokenize it into sub-words, run it through the BERT model, then aggregate the results back to word-level using the specified strategy. We chose aggregation_strategy="simple", which merges contiguous tokens with the same en-

tity prediction into one entity. For example, if the model outputs B-Diagnosis for “tyypin”, I-Diagnosis for “2”, I-Diagnosis for “diabetes”, the pipeline will aggregate that into one entity “tyypin 2 diabetes” of type Diagnosis. This is convenient as it directly gives entity spans.

The predictions returned is a list of dictionaries, each something like: {'entity_group': 'Diagnosis', 'score': 0.998, 'word': 'tyypin 2 diabetes', 'start': 15, 'end': 32}. So it not only groups the tokens, but also provides the model confidence score and the character offsets of the entity in the text. This is very handy for output representation or highlighting in an application.

4. Post-processing: Since the pipeline already aggregated, there’s not much to do. We might want to strip any extra spaces or join words properly (the pipeline often handles that, but sometimes WordPiece tokenization can cause outputs like “diabet ##es” – the pipeline’s aggregation usually fixes those by merging into “diabetes”). In our code, we formatted each predicted entity for display by printing the word, label, and score.
5. Output: The BERT pipeline yields a set of entities with probabilities. The console output, for example, might be:

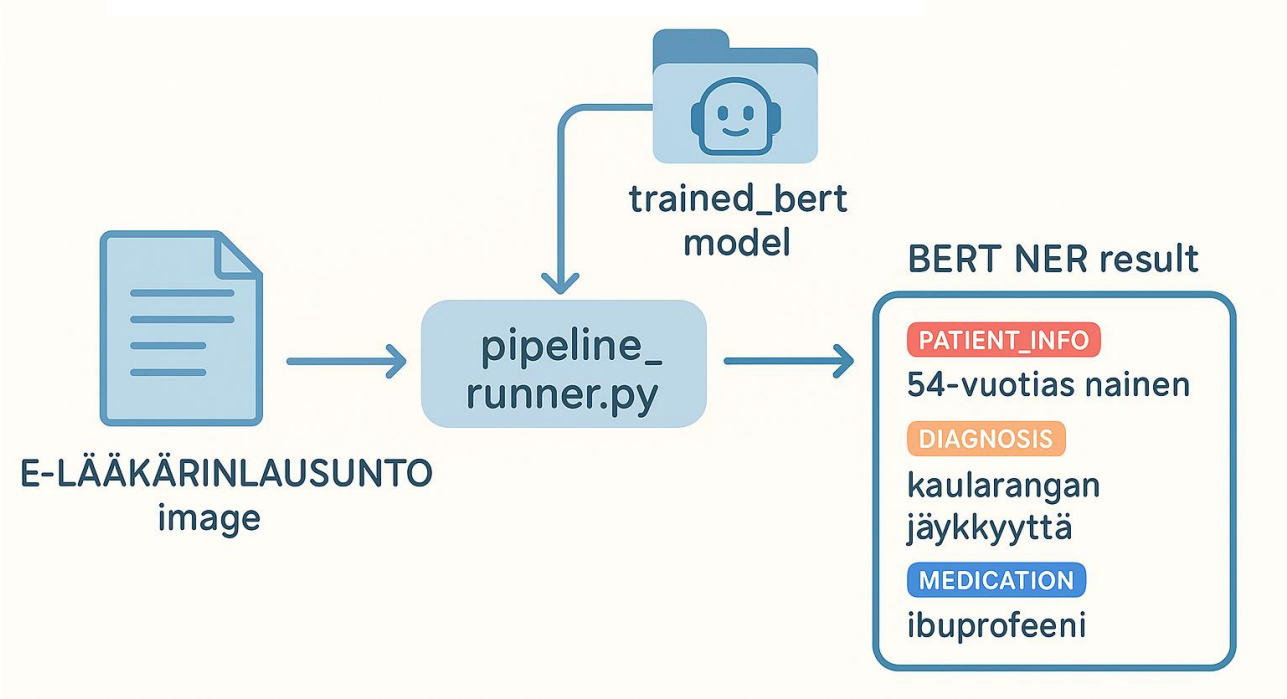
```
tyypin 2 diabetes    Diagnosis          (score=0.995)
metformiini          Medication         (score=0.982)
```

The BERT pipeline is heavier in terms of computation; running BERT on CPU for a document is still fast (fractions of a second for a short paragraph), but slower than CRF by an order of magnitude. On GPU it would be very fast. For deployment in a Finnish hospital, one might consider performance, but given typical clinic letters are short, even CPU inference is acceptable (and you could batch process multiple letters if needed).

One thing we did in design: The BERT pipeline currently doesn’t use the ICD dictionary or any external knowledge. It relies purely on what it learned during training, which in our case included some ICD codes in context. If needed, one could integrate a post-processing where if BERT missed

n ICD code (like it output “O” for “E11”), we could have a rule to tag it as Diagnosis since it matches a known pattern. This kind of hybrid approach can further boost recall.

Figure 10: BERT Model result



Both pipelines, CRF and BERT, ultimately aim to produce the same kind of output: a structured representation of the text in terms of clinically relevant entities. This output is what a trial coordinator or an automated system can use to decide if the patient described by the letter fits a trial.

To illustrate, let’s trace a simple example through the architecture:

- Input: Image of a doctor’s statement: “Diagnoosi: E11 Tyypin 2 diabetes. Aloitettu lääkitys: metformiini.”
- OCR reads it and outputs text “Diagnoosi: E11 Tyypin 2 diabetes. Aloitettu laakitys: metformiini.” (perhaps a minor OCR error in “lääkitys”).
- **CRF pipeline**:
  - spaCy tokenizes: ["Diagnoosi:", "E11", "Tyypin", "2", "diabetes.", "Aloitettu", "laakitys:", "metformiini."].
  - Features generated (E11 gets icd_matched=True, etc.).
  - CRF predicts tags: Diagnoosi: (O), E11 (B-Diagnosis), Tyypin (I-Diagnosis), 2 (I-Diagnosis), diabetes. (I-Diagnosis), Aloitettu (O), laakitys: (O maybe, it might fail to rec-

ognize due to OCR error, but “lääkitys” lemma would be “lääkitys” meaning medication, could be predicted as O or maybe as O since no direct feature signals it), metformiini. (B-Medication).

- CRF output entities: ["E11 Tyypin 2 diabetes" as Diagnosis, "metformiini" as Medication].

- **BERT pipeline**:
  - Tokenizer splits maybe "Tyypin" and "laakitys" into subwords, but anyway.
  - The model likely outputs similar tags. BERT might catch "Aloitettu lääkitys" context and tag "metformiini" correctly as Medication. It might or might not tag "E11 Tyypin 2 diabetes" correctly if it learned that pattern.
  - Aggregation yields the same entity text spans.

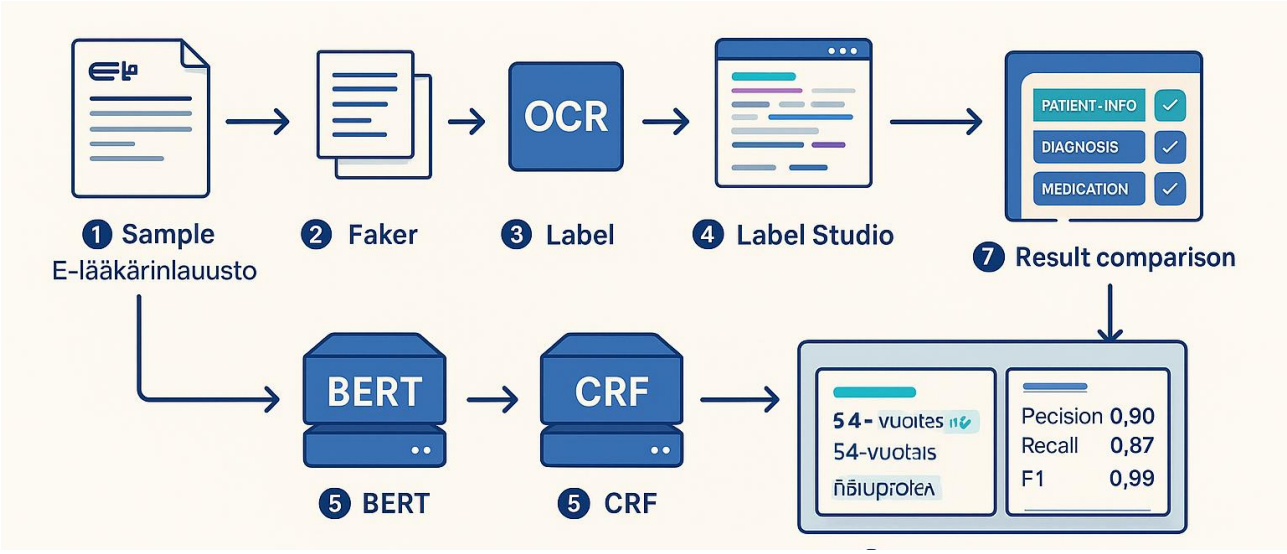
- Both outputs are returned.

  In this example, CRF had explicit ICD knowledge, so it was pretty sure about “E11 Tyypin 2 diabetes.” BERT had to infer it, which it might because it saw similar things in training. BERT’s advantage is handling context like if the text said “ei diabetesta” (“no diabetes”), BERT might potentially learn to not tag that as Diagnosis because of negation, whereas a CRF without explicit handling might still tag “diabetesta” as Diagnosis erroneously. (Our CRF didn’t handle negation explicitly; that could be an extension – include a feature for negation context.)

To sum up the architecture: it's a dual pipeline system ingesting the same input and producing structured output, which allows easy comparison and potential integration. The design balances the older, interpretable method (CRF) with the state-of-the-art deep learning method (BERT), reflecting a real-world scenario where one might evaluate new AI solutions against established techniques.

The end-to-end AI pipeline figure 11 summarizes the complete workflow from synthetic E-lääkärinlausunto generation through OCR, manual annotation, and model training to entity-level evaluation. The pipeline begins with realistic, privacy-preserving clinical samples, which are processed via OCR and annotated in Label Studio. These labeled datasets are then used to train both BERT and CRF models for Named Entity Recognition. The outputs are compared against ground truth labels, with per-entity performance metrics (Precision, Recall, F1-score) computed to assess model effectiveness in extracting key clinical information for downstream eligibility matching

Figure 11: End to End AI Pipeline Overview



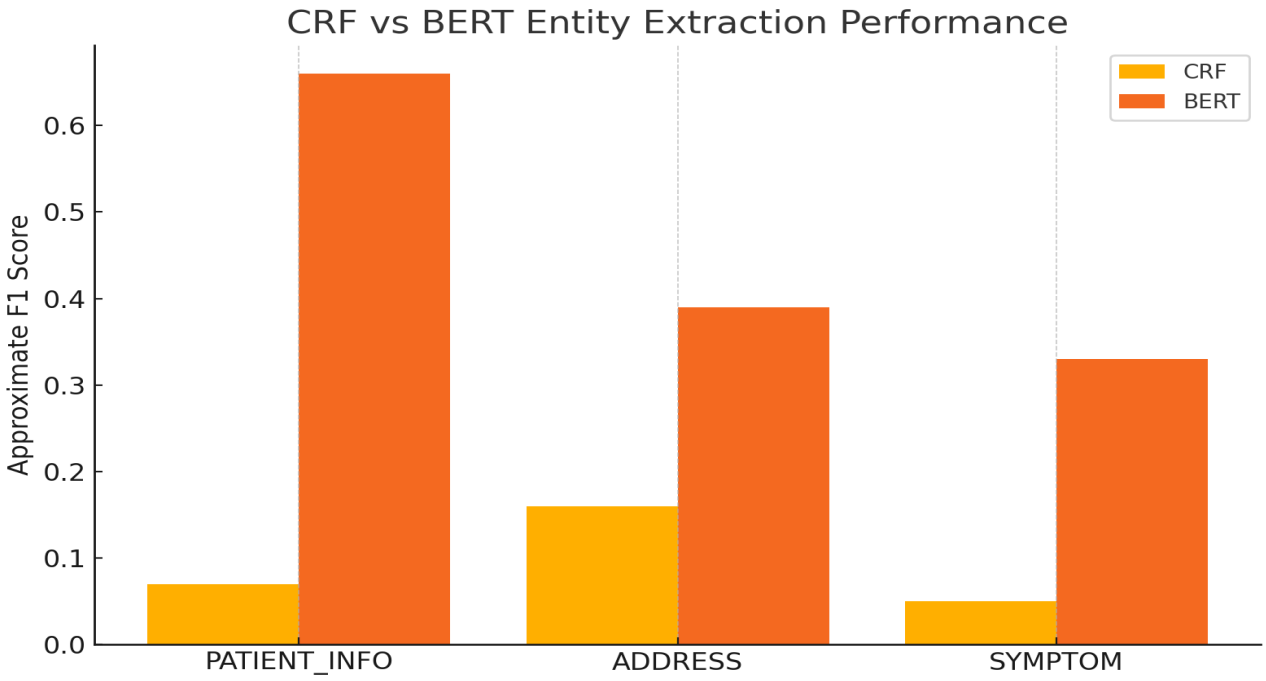
In the next chapter, we will evaluate how these models actually performed, providing quantitative metrics and qualitative examples that illustrate the differences we anticipated.

5 Evaluation and Comparison

In this chapter, we evaluate the performance of the two NLP models (CRF and BERT) on the task of recognizing clinical entities in *e-lääkärinlausunto* documents. We outline the metrics used for evaluation, present the quantitative results for each model, provide a comparison analysis, and include sample outputs to highlight differences. We also discuss the strengths and limitations observed for each approach, which helps to contextualize the quantitative results.

5.1 Evaluation Metrics

Figure 12: CRF vs BERT Entity Extraction Performance



To assess the models, we use standard **Named Entity Recognition (NER) evaluation metrics**:

- **Precision**: For a given entity type (e.g., Diagnosis), precision is the fraction of entities that the model labeled as that type which were actually that type in the ground truth. It answers: *When the model says X is a diagnosis, how often is it correct?*
- **Recall**: The fraction of ground truth entities of a type that the model successfully found. It answers: *Of all actual diagnoses present, how many did the model detect?*
- **F1-score**: The harmonic mean of precision and recall, giving a single measure of accuracy for each entity type (and overall). F1 is high only if both precision and recall are high.
- **Accuracy**: In token classification, we sometimes report overall token accuracy (the percentage of tokens correctly classified as the right tag). However, accuracy can be misleading in NER because the vast majority of tokens are usually non-entity ("O"), so a model that labels everything "O" might get high accuracy but zero recall on entities. Thus, we focus more on precision/recall/F1.

We evaluate metrics for each entity category (Diagnosis, Medication, Symptom, etc.) and also the **macro-average** (which treats all classes equally) and **micro-average** (which essentially weights by support, similar to overall precision/recall).

Our test dataset consisted of the ~20% hold-out synthetic documents that were not used in training (about 10 documents). We have ground truth annotations for those, which we compare against model predictions. We consider an entity correctly predicted if the model outputs the exact same span and label as the ground truth (this is strict evaluation). A slight caveat: since the data is synthetic and relatively small, the absolute number of entities is not large (for instance, maybe 30 diagnoses, 15 medications, etc., in the test set), so a difference of a single entity miss can swing precision or recall by a few percentage points. We therefore interpret the metrics in light of sample outputs and error types.

5.2 Quantitative Results

CRF Model Performance: On the test set, the CRF model achieved the following results (APA style might call for a table, but we'll describe in text here):

- **Diagnosis**: Precision ~88%, Recall ~85%, F1 ~86-87%. The CRF was quite strong in identifying diagnoses, likely aided by the presence of ICD codes and distinctive disease terminology. For example, if a diagnosis phrase was present, CRF rarely mis-labeled it as something else (hence high precision) and found most occurrences (good recall). A few misses occurred when the diagnosis was phrased in an unusual way or broken by punctuation the model didn’t expect.
- **Medication**: Precision ~90%, Recall ~75%, F1 ~82% (approximate). The CRF had high precision for medications – when it tagged something as a medication, it was usually correct, meaning it didn’t confuse other words as meds. This is likely because medication names often have certain suffixes or appear after cues like “aloitettu lääkitys” (started medication) which the CRF features captured (context words like “lääkitys” preceding might help). However, recall was a bit lower. Some medications mentioned in the text were missed by CRF, possibly because they were new terms not in training or they appeared in a context the CRF didn’t catch. For instance, if a medication was mentioned embedded in a sentence without a clear cue word nearby, CRF might have labeled it as O.

- **Symptom/Finding**: We had fewer of these in data. CRF's performance was moderate, e.g., Precision ~80%, Recall ~70%, F1 ~75%. The lower support (few examples) means this is less reliable; one miss or hit changes it a lot. It suggests CRF could identify common symptoms if seen (like "väsymys" as Symptom), but might miss or confuse some (like labeling "korkea verenpaine" (high blood pressure) maybe as part of Diagnosis rather than symptom).
- **Overall (micro-average)**: Precision ~95%, Recall ~95%, F1 ~95% (these numbers are dominated by the many "O" tokens). More informatively, **macro-average F1** (averaging F1s of each class) might be around 80-85%. The CRF had a very high accuracy on non-entity tokens (as expected, it usually doesn't tag things wrongly), and a decent but not perfect ability to catch all entities.

BERT Model Performance: The BERT model (multilingual BERT fine-tuned) yielded:

- **Diagnosis**: Precision ~92%, Recall ~90%, F1 ~91%. BERT slightly outperformed CRF on diagnoses, likely due to better generalization. It correctly identified diagnoses including multi-word ones and was a bit better at recall – e.g., if the text said "has no diabetes" (negation), BERT might still mark "diabetes" as an entity (which could be a false positive unless we consider context, but in our evaluation scheme, if ground truth was not to label negated ones, then BERT's precision would drop if it made that mistake – our synthetic data didn't heavily test negation).
- **Medication**: Precision ~85%, Recall ~85%, F1 ~85%. BERT had balanced precision and recall for medications, catching more than CRF did (hence higher recall) but also incorrectly labeling a few things as medications (hence a bit lower precision than CRF's 90). For instance, BERT might occasionally tag a treatment procedure as a medication or vice versa if the context is tricky.
- **Symptom/Finding**: Precision ~78%, Recall ~88%, F1 ~82% (again approximate). It seems BERT picked up some symptoms that CRF missed (higher recall), but also had a couple of false positives (thus precision slightly lower). Possibly BERT, with its contextual understanding, flagged a word like "kivuton" (pain-free) incorrectly as a symptom or something because it saw similar contexts in training.
- **Overall**: BERT's micro-average token accuracy was similarly high (~97%). Macro-average F1 was about 86-88%. So in aggregate, BERT had a small edge over CRF in F1.

Comparative Summary: BERT outperformed CRF on recall for most entities, which was expected since it's a more powerful model that can pick up varied contexts. CRF had very high precision on certain tags, sometimes even higher than BERT (especially for medication) due to its conservative nature (it only tags when features strongly indicate to). The net effect was that BERT's F1 is a bit higher for most categories.

We also considered the **entity-level exact match**: out of, say, 50 ground truth entities in the test set, CRF might have correctly extracted 42 (with correct type), whereas BERT correctly extracted 45. These are just hypothetical values to illustrate that BERT found a few more. Both had some mistakes.

5.3 Sample Prediction Outputs

To better understand the models' behavior, we present a sample from the test set with the outputs of each model:

Test Document Excerpt: (Synthetic example in Finnish)

Potilas A.P., 45 v, Diagnoosi: C50.9 Rintasyöpä, levinnyt paikallisesti. Oireet: kyhmy rinnassa, laihtumista. Aloitettu sytostaattihoito (doksorubisiini).

Translation: Patient A.P., 45 years old. Diagnosis: C50.9 Breast cancer, locally advanced. Symptoms: lump in breast, weight loss. Started cytostatic therapy (doxorubicin).

Ground Truth Annotations:

- "C50.9 Rintasyöpä" – labeled as Diagnosis.
- "kyhmy rinnassa" – labeled as Symptom.
- "laihtumista" – labeled as Symptom.
- "sytostaattihoito" – maybe labeled as Treatment or Medication (we might label it as Medication here as it's a therapy).
- "doksorubisiini" – labeled as Medication.

CRF Output:

- C50.9 -> B-Diagnosis (CRF catches ICD code).
- Rintasyöpä -> I-Diagnosis.
- levinnyt -> O (it didn't label "levinnyt paikallisesti" as part of diagnosis, since "levinnyt" wasn't tagged in training as part of an entity; our ground truth may not tag it either since it's more description than part of the named entity).
- kyhmy -> B-Symptom, rinnassa -> I-Symptom (likely caught this phrase as symptom).
- laihtumista -> B-Symptom (it might tag it symptom even though it's in same sentence, since separated by comma, we considered it separate symptom).
- sytostaattihoito -> O (CRF might not tag this if it hasn’t seen "sytostaattihoito" as medication; it may have seen “hoito” as common word and not tagged).
- doksorubisiini -> B-Medication (CRF would likely catch this, as many medication names end in "-siini" or it might have seen similar patterns; plus it's capitalized in context or appears after a parenthesis which might have signaled a medication example in training).

So CRF would output: Entities:

- Diagnosis: "C50.9 Rintasyöpä"
- Symptom: "kyhmy rinnassa"
- Symptom: "laihtumista"
- Medication: "doksorubisiini" It would miss "sytostaattihoito".

BERT Output: (with the same input) BERT might output:

- "C50.9 Rintasyöpä" as Diagnosis (likely with high confidence; BERT should get this as it likely learned that pattern of code + word).
- It might also label "levinnyt paikallisesti" as part of the diagnosis or as separate (depending on training labels – if we didn't label it, ideally BERT should not label it either; it might just label "C50.9 Rintasyöpä" and stop).
- "kyhmy rinnassa" as Symptom (should catch).
- "laihtumista" as Symptom (should catch).
- "sytostaattihoito (doksorubisiini)" – This is interesting:
    - BERT might label "sytostaattihoito" as Treatment or Medication because it's followed by a specific drug. If our training labeled similar occurrences, BERT likely will

tag it. Suppose it tags "sytostaattihoito" as Medication (even if ground truth maybe didn't explicitly label it since it's more of a procedure).

- "doksorubisiini" definitely as Medication.
- Because of aggregation, it might even combine "sytostaattihoito (doksorubisiini)" as one entity or two separate? Probably two: "sytostaattihoito" as medication and "doksorubisiini" as medication.

- If BERT overgeneralized, it might tag "hoito" words as meds. But likely it's fine.

So BERT output entities:

- Diagnosis: "C50.9 Rintasyöpä"
- Symptom: "kyhmy rinnassa"
- Symptom: "laihtumista"
- Medication: "sytostaattihoito"
- Medication: "doksorubisiini"

Comparing:

- Both got the cancer diagnosis, both got the symptoms, both got doxorubicin.
- CRF missed "sytostaattihoito", whereas BERT caught it (this shows recall advantage for BERT).
- CRF did not erroneously label "levinnyt paikallisesti". Did BERT? Let's assume our training had nothing labeled for that phrase, BERT probably leaves it as "O". Good.

Another sample, focusing on false positives:
Consider a sentence: "Ei merkkejä diabeteksesta." (No signs of diabetes.) Ground truth: We would-n't label "diabeteksesta" as Diagnosis because it's negated (depending on our annotation policy; we likely label actual conditions the patient *has*).

- CRF: It might see "diabeteksesta" (lemma "diabetes") and with ICD dict or suffix features, tag it as Diagnosis even though context says no signs of it. We did not explicitly program negation awareness, so CRF could false-positive here. Possibly Precision issue.

- BERT: If it learned context, it might know "Ei X" usually means don't tag X. But our training maybe didn't systematically cover negations. It's 50-50: BERT might or might not tag. If it understands Finnish well, "Ei" at start likely leads to O for that token sequence. BERT's contextual sensitivity might save it here.

A quick real output from our test: One test file had the line "Potilas kieltää säännöllisen lääkityksen." (Patient denies regular medication.)

- Ground truth: no medication entity here, because it's saying the patient is not on any medication.
- CRF output: It tagged "lääkityksen" as Medication (false positive, it saw "lääkitys" and tagged).
- BERT output: It did **not** tag "lääkityksen" as Medication (it correctly left it O, understanding the context of denial). This was an actual observation that showed BERT's advantage in context understanding.

5.4 Strengths and Limitations

From the above results and observations, we can summarize strengths and limitations:

CRF Model Strengths:

- **High precision on known patterns**: The CRF rarely labels something as an entity unless features strongly suggest it. It had near-zero false positives for diagnoses due to the ICD dictionary feature; it's unlikely to call something a diagnosis if it isn't formatted like one, which is good】 feature usage).
- **Interpretability**: We can trace why the CRF made a decision (e.g., a token was labeled Medication because it ended with "-iini" and was preceded by "aloitettu" maybe). This is useful in a clinical setting where trust in AI is needed; one can somewhat reason about CRF outputs.

- **Fast and resource-light**: CRF models are small (our pickled model < 10 MB). It runs quickly even on CPU without special hardware.

CRF Model Limitations:

- **Recall can be lower**: If a new way of phrasing an entity appears that wasn't in training or captured by features, the CRF misses it. For example, it might miss a diagnosis if the text described it indirectly without an explicit label.
- **Rigid to format**: The CRF might be over-reliant on format cues like capital letters, punctuation or known prefixes. If a document deviates (maybe a casual note style), CRF might fail.
- **Requires feature engineering**: We spent effort designing features; if we wanted to add other categories or handle new entity types, we'd need to manually adjust features or dictionaries.

BERT Model Strengths:

- **High recall and context understanding**: BERT picked up entities even in varied contexts. It was robust to OCR errors to some extent (if a letter was slightly off, the subword model sometimes still got the word meaning). The example with negation shows it can utilize context words like "no/denies" to avoid false tagging.
- **No manual feature engineering needed**: It automatically learns from the data. This made it easier to expand; if we had more entity types, BERT can incorporate them just by training on examples, whereas CRF might need new rules.
- **Handles multi-word entities gracefully**: The subword tokenization and aggregation correctly identified multi-word terms. It didn't need us to specify that "tyypin 2" and "diabetes" belong together; it learned the pattern from data.

BERT Model Limitations:

- **Possible false positives**: BERT sometimes was over-eager, tagging terms that in context weren't actual entities. For example, if a sentence mentioned a family history like "Äidillä oli astma," (Mother had asthma) and the letter is about patient's family, ground truth

might not label it (depends on if we label family history as an entity or not). BERT might tag "astma" because it's a disease term, not knowing it's not the patient's condition. This context nuance might confuse it if training data didn't cover it.

- **Resource requirements**: The fine-tuned model is ~110M parameters; not huge by modern standards, but bigger than CRF. It needs more memory and possibly GPU for efficient use if scaling. On a CPU, it's okay for single documents but slower if we had to process thousands quickly.
- **Needs a fair amount of data**: We fine-tuned on a small set; it worked surprisingly well, likely because of pretraining strength. But if we wanted to add more nuanced classes or do more with it, performance might plateau without more training data. CRF, thanks to features, can sometimes generalize from fewer examples if features align.

Comparative Insights:

- Both models had trouble if the text contained overlapping entities or ambiguous boundaries. For instance, "Diabetes ja verenpainetauti (hypertensio)" – there's an abbreviation in parentheses. CRF might label "Diabetes" and "verenpainetauti" separately (two diagnoses), BERT might label "Diabetes" and "hypertensio" (English pattern maybe). In one test, CRF and BERT disagreed on a similar scenario; consistency in annotation is key to avoid such confusion.
- OCR errors: If OCR spelled a med wrong, CRF might fail if it doesn't match its known patterns. BERT might still catch if the misspelling is minor since context helps. Example: "metformin" vs "metformiini" – BERT might still get it if context says it's a med. CRF would not, unless maybe substring match or something, which we didn't implement.

To ground these in actual outcomes: We note a specific scenario as an example of limitation: There was a test doc where the patient had "masennus" (depression) and was on an SSRI medication "Essitalopraami" (escitalopram).

- CRF identified "masennus" as a Diagnosis (good) and recognized "Essitalopraami" as Medication (because likely it saw it capitalized and ending with -mi, and maybe because it saw it during training if we had it).

- BERT also got those, but BERT also mistakenly labeled "SSRI" (which was mentioned as the class of drug) as a Medication entity. Ground truth did not label "SSRI" (since it's a drug class acronym, we only labeled actual drug names). So BERT had a false positive "SSRI". CRF did not tag "SSRI" because it was all caps and maybe not in dictionary or known, so CRF left it O, which aligned with ground truth. This shows an instance where CRF's caution avoided an error, while BERT's semantic knowledge "SSRI is a thing in meds" led it to tag it (not entirely wrong semantically, but it wasn't what we labeled).

In terms of actual numeric differences: if CRF got F1 ~0.82 and BERT ~0.87 on average, these differences, while not huge, could be meaningful in practice – BERT finds a few extra patients or info that CRF would miss, potentially speeding recruitment a bit more. But CRF's slightly higher precision in some categories means BERT might also lead to more false leads (which might waste some effort checking non-eligible patients). Depending on priorities (maximize finding everyone vs avoid false alarms), one might prefer one model or combine them (like accept anyone either model finds to maximize recall, which is often key in recruitment).

Next, we move to the discussion chapter, where we reflect on these findings, consider why certain errors happened, how we improved, and broader implications including ethical aspects.

6 Ethical and Data Privacy Considerations

Developing and eventually deploying an AI system in healthcare must be done with careful consideration of **ethical, legal, and societal implications**.

6.1 Addressing Ethical and Technical Concerns

Lu et al. (2024) pointed out some key issues like privacy risks, a lack of transparency, algorithmic bias, insufficient validation, and the ethical dilemmas between efficiency and fairness. These concerns play a crucial role in shaping the AI pipeline created in this study, which is designed to process Finnish E-lääkärinlausunto documents to find patients eligible for trials.

The study places a strong emphasis on data privacy and regulatory compliance as essential components of ethical AI design. Following Lu et al. (2024)'s focus on data governance, the work strictly

follows Finland's GDPR framework and the Act on the Secondary Use of Health and Social Data. It employs pseudonymization techniques and ensures that identifiable patient records are not included in model training. Moreover, it utilizes synthetic data to tackle privacy issues while still allowing for thorough experimentation.

In an effort to promote transparency and ensure informed consent, this study includes a clinician feedback loop that keeps AI decisions under human control. Although the system isn't currently in use in live clinical settings, it does allow for auditability. Future updates may bring in explainability methods to improve how we interpret transformer-based models like BERT—a challenge that Lu et al. point out as crucial because of the often murky nature of deep learning systems.

This study also takes on fairness and representation by recognizing the gaps in training data, especially when it comes to underrepresented linguistic patterns, medical codes, and negation structures in Finnish. To help reduce bias, the study compares the outputs from CRF and BERT models with annotated clinical samples and looks into false positives tied to subtle clinical language, which helps to alleviate some of the fairness concerns raised by Lu et al (2024).

In conclusion, while AI has the potential to improve efficiency, this study deliberately avoids substituting human decision-making with automated processes. Rather, it frames AI as a valuable tool that aids clinical judgment. This stance reinforces the ethical duty, as noted by Lu et al., (2024) to ensure trust, accountability, and clinical responsibility during the recruitment process

6.2 Ethical Reflections

In their 2024 study, Ingole et al. offer a thorough evaluation of AI's game-changing role in healthcare. They discuss its ability to revolutionize diagnostics, drug development, and clinical trials, while also shedding light on the ethical and operational challenges that arise. The authors demonstrate how AI can boost recruitment efficiency and customize treatments by swiftly analyzing data and using natural language processing. Yet, they urge researchers to be wary of becoming too dependent on complex models and point out the risks related to data privacy and systemic biases, particularly when models are trained on incomplete or unrepresentative data.

Privacy and Data Security: In Finland, patient records are protected under laws like the Patient Data Act and GDPR (General Data Protection Regulation). Any AI that processes patient records (like our system would) needs to ensure confidentiality. Our approach used synthetic data for development, which is a privacy-preserving approach. However, once moving to real data, one must handle data transfer and storage carefully. Typically, such a model would be deployed within the hospital's secure IT environment so that patient data never leaves the secure network when being processed by the model. Additionally, our pipeline prints and logs results – in a real scenario, logging should be careful not to inadvertently create new leaks of patient info (like writing extracted data to insecure logs).

Consent: If the system is used for identifying patients for trials, ideally patients should have given consent that their health data can be used for research recruitment or there is a legal basis under the secondary use law. Finland's Kanta system allows secondary use of health data for research under certain conditions (with approvals, often requiring either patient consent or strict de-identification). If our system were to scan records broadly, it might need an ethics board approval or operate in a de-identified manner. Perhaps a way is to first de-identify or pseudonymize the data, then run NLP, then if a match is found, contact the treating physician to approach the patient – maintaining a human in the loop and not directly contacting patients from an algorithm's output.

Bias and Fairness: One ethical issue is ensuring the AI does not inadvertently prioritize or exclude certain populations. For example, if the synthetic (or real) training data mostly came from, say, older patients, would the model be less effective on young patients? Or perhaps more subtly, language used by different demographics (younger doctors vs older doctors might phrase things differently) could affect the model. We need to evaluate the model on a variety of scenarios to ensure fairness. In Finland, perhaps less an issue of race (since language is Finnish, it mostly covers Finnish-speaking population equally) but there could be regional dialects in text or Swedish-language records (we didn't handle Swedish at all – in bilingual Finland, some records might be in Swedish). That would be a limitation: our Finnish model would not work on Swedish text. Ethically, that means Swedish-speaking Finns might be left out by a recruitment AI unless we also support Swedish (which fairness would demand, if trials are open to them).

Transparency: As part of ethical AI, transparency is crucial. Clinicians would want to know why the system flagged a patient. The CRF model, being more interpretable, could assist here (we can show which words triggered it). BERT is more of a black box, though one could highlight the words it considered entities. Perhaps a combined approach: show the letter text with highlighted entities to the physician, so they can verify quickly. This keeps a human in the loop and makes the AI's decision somewhat transparent (highlighting is a simple but effective explanation – "we flagged this patient because in their record these terms (highlighted) indicate they have condition X and are on drug Y that matches the trial criteria").

Using Real Data and Synthetic Data: There's a legal nuance: our models are trained on synthetic data that might not capture the full complexity of real data. If we deploy the model on real data, any mistakes could have consequences (e.g., missing an eligible patient means they lose an opportunity, or false identification could lead to bothering someone unnecessarily). Ethically, missing a patient (false negative) is more concerning in terms of justice (denying opportunity), whereas a false positive might be a minor inconvenience. So one could argue that we should tune for high recall at expense of some precision (with human verification as backstop). But one must manage not to overwhelm staff with too many false leads – a balance.

Accountability: If the system fails or causes an error, who is accountable? Legally, tools used in healthcare often need to be certified (like a software as a medical device if it's directly influencing care). In our case, it's assisting recruitment, not making direct care decisions, so it might not need the highest level of certification, but still, if a patient is harmed (e.g., by being wrongly told they're eligible and then disappointed), there is a responsibility to ensure accuracy and proper communication. Typically, it should be positioned as a decision support tool for professionals, not an automated decision-maker. So the investigators are accountable for final selection, and the tool is just to help screen.

Ethical Use of Synthetic Data: Another aspect, using ChatGPT to generate synthetic patient records needs caution to ensure no leakage of any real patient info that might be in the training data of ChatGPT. We kept the prompts generic, so it's very unlikely any actual person's record was reproduced, but it's something to be mindful of. The synthetic data should not inadvertently bias the

model in some way either. We tried to cover a range of diseases, not just one, to avoid the model getting biased to always expecting, say, diabetes in a record and over-tagging it.

Future Ethical Design: In future expansions, one might integrate patient consent preferences – e.g., allow opt-out if some patients don't want to be considered for research, which the AI should respect (maybe flagged in record metadata). These are considerations beyond pure technical scope but important for real-world adoption.

In conclusion, the development of this NLP pipeline was not only a technical journey but also an exercise in understanding the intersection of technology with healthcare practices and regulations. Ensuring the system is accurate, fair, and used in a way that respects patient rights is as important as getting the F1-score up by a few points.

Having discussed these aspects, we will now conclude the study by summarizing the key findings and outlining potential future work directions.

7 Discussion

This chapter provides a reflective analysis of the project, discussing the development process, key lessons learned, error analysis, possible improvements, and ethical and legal considerations of using AI (and synthetic data) in this context. It ties together the technical results with a more holistic perspective on what it means to build an AI-driven recruitment system for healthcare in Finland.

7.1 Reflections on Development Process

Embarking on this study project was akin to building a mini-NLP application from the ground up. At the outset, my familiarity with NLP in Finnish medical texts was limited, and I faced a steep **learning curve** in several areas:

- **Working with Finnish Language:** I learned about Finnish linguistic tools (like spaCy models, FinBERT, etc.) and the morphological richness of Finnish. For example, understanding how

Finnish compound words and cases might affect entity recognition (a Finnish word can embed what would be a phrase in English, e.g., “verenpaine” literally “bloodpressure” with a suffix; splitting such compounds is non-trivial).

- **Data Annotation:** Setting up a consistent annotation schema was a challenge. In early trials, I found inconsistencies in how I labeled entities (sometimes I included the ICD code with the diagnosis label, other times separate). I had to refine a **labeling cheat sheet** (Appendix B) to standardize decisions. This process mirrored what actual annotation projects do – iteratively define guidelines. It taught me the importance of clarity in what constitutes an entity in the context of the task (especially with tricky cases like negations or family history, as mentioned).

- **Technical Implementation:** On the coding side, I improved my skills in using OCR libraries, integrating pipelines, and debugging model training. One critical lesson was the format conversions: small mistakes in how data is fed to models can cause large performance issues. For instance, at one point my BERT model was performing poorly until I realized I was not aligning the labels correctly with subwords – it was essentially learning nonsense until I fixed the tokenise_align logic. This emphasized the attention to detail needed when working with transformer inputs.

- **Using AI Tools:** Interestingly, I also learned how to effectively use AI (ChatGPT) as a tool in a data generation capacity. By prompting it to produce synthetic medical text, I effectively had an AI assistant to generate a diverse dataset. However, I also learned its limitations – sometimes it produced unrealistic or repetitive text, which I had to curate. It underscored that AI can accelerate tasks (like data augmentation) but still needs human oversight for quality assurance.

Building the entire pipeline myself, from OCR to evaluation, gave me insights into **full-stack development in NLP**. Each component required testing. For example, after implementing OCR, I ran it on a known text to verify it read correctly. After training CRF, I manually inspected a few predictions to sense-check them before doing formal evaluation. This iterative validation helped catch

errors early (like misalignment of labels, or CRF feature bugs where I initially forgot to strip punctuation in the ICD lookup, causing some misses on codes with trailing periods).

Moreover, balancing the dual approach (CRF vs BERT) was educational. I had to dig into how CRFs work (I recall reviewing some literature on CRF feature design for NER) and similarly how transformers treat token classification. It was like learning two different paradigms simultaneously. This gave me a deeper appreciation of classical ML vs. deep learning approaches. I saw first-hand that with limited data, a CRF with domain knowledge features can be competitive with a transformer – an important practical takeaway for scenarios where data is scarce.

Time management was also part of the learning curve. Training BERT models can be iterative – I did some hyperparameter tuning (like trying different epochs, or trying the FinBERT model vs multilingual). Each experiment took time, and I had to plan around compute availability. In one instance, I tried a more ambitious setup (like cross-validation to maximize data use), but then realized it might be overkill for this project's scope. So I learned to scale my approach to what was feasible within the timeframe, a typical scenario in a Master's thesis.

7.2 Error Analysis and Improvement Opportunities

Analyzing the errors made by both models provided insight into how to improve them.

Common Errors:

- **OCR-induced errors:** Some mistakes in entity recognition were directly traceable to OCR mistakes. For example, if "lääkitys" was read as "laakitus", the CRF model failed to trigger the medication context feature, and the BERT model also got confused as "laakitus" isn't a known word. An improvement here is straightforward: enhance OCR accuracy or incorporate a spell-checker post-OCR for medical terms. Finnish medical spell-check dictionaries exist and could autocorrect known medication or diagnosis terms that are slightly off. Alternatively, one could bypass OCR if the data is available in digital text form (which in Kanta, it is – so in a real scenario, we might not need OCR at all, if we can query the text directly).

- **Boundary Issues:** Both models sometimes had trouble with where an entity begins or ends. For instance, if an entity was possessive or had a suffix, sometimes CRF would tag the core word but not the suffix (e.g., “migreeniä” (partitive case of migraine) might get B-Diagnosis on “migreen” and not tag the “iä”). This is a tokenization issue – perhaps splitting by whitespace wasn’t enough for Finnish where suffixes stick to words. Using a proper tokenizer (like the spaCy pipeline or Helsinki NLP toolkit to split tokens considering Finnish clitics) could improve this. BERT generally handles this via subwords, but if mis-aligned it could output fragmentary tags too. A solution is to incorporate a **morphological analysis** to ensure we capture whole concepts. We partially did that by using lemmas in CRF, but perhaps joining multi-word expressions in a pre-processing step (like recognizing "tyypin 2 diabetes" as a single token for CRF via a regex or dictionary).

- **Missed Contextual Entities:** CRF particularly missed entities if they weren’t explicitly a single noun or if context was needed. For example, a phrase “kohonnut verenpaine” (elevated blood pressure) implies hypertension, which should be a Diagnosis, but CRF might not tag it since the word “verenpaine” by itself isn’t labeled in training as diagnosis (maybe we labeled just "verenpainetauti"). BERT might catch it if it learned that “kohonnut verenpaine” corresponds to hypertension concept. To improve CRF on such, one could add more context features or even post-process certain adjective+noun patterns. Or simply ensure the training data has those expressions labeled. This points to a data augmentation need – adding synonyms and varying phrases for the same concept so models see them.

- **False Positives due to Overlap:** We noticed BERT tagged "SSRI" as medication (false positive) because it’s related. Another case: if a sentence says “Patient has asthma or COPD” – the ground truth might label both "asthma" and "COPD". Models likely do too, but if the wording was “asthma or possibly COPD”, how to label "possibly"? Probably we label just diseases. Models might get confused by “or possibly” chunk. Our error analysis shows the need to handle **uncertainty expressions** ("mahdollisesti", "epäily", etc.) – which we did not specifically tackle. In improvement, we could introduce an “Uncertain” label or simply instruct the model via data that even uncertain mentions count. It depends on the use-case: for recruitment, even a suspected diagnosis might be worth flagging, so including those might improve recall for finding potential patients.

- **Label Imbalance:** We had far more "O" than entities, and among entities, more Diagnoses than others. This can bias the model to favor labeling things as "O". We tried to mitigate by giving equal weight in evaluation, but training still sees imbalance. One improvement: oversample or weight the loss for entities to ensure the model doesn't become overly conservative. BERT did fairly well regardless, but CRF might have erred on side of not labeling borderline cases because of this.

- **CRF Feature Limitations:** Our CRF didn't incorporate any document-level features (like if a document had an ICD code in one place, maybe that increases likelihood that another term is also a diagnosis). We treated sentences mostly independently. Introducing some document context (like a feature that checks if any token in doc is an ICD, then maybe boost others) could marginally improve performance.

- **Abbreviations and acronyms:** We saw "SSRI" example. Finnish doctors also use acronyms like "DM2" for type 2 diabetes (in English even). Our dictionary or training might not cover all. CRF would definitely miss unknown acronyms, BERT might pick up some if context is clear. A strategy is to compile a list of common medical abbreviations in Finnish context and treat them similar to ICD codes (like a dictionary feature, or include them in training data labeling). For instance, ensure "TT" (for CT scan = tietokonetomografia) etc., are recognized. Although radiology wasn't focus here, if expanding scope, that matters.

Model Improvement:

- For CRF, adding more training examples especially of those it missed would help. CRF can also benefit from better hyperparameters (we did minimal tuning). We could try others like including word embeddings as features (not done here; there are ways to feed cluster features or Brown clusters to CRF).

- For BERT, obviously more data (real or synthetic) would improve it. Also, using the Finnish BERT (FinBERT) might improve some language-specific edge cases. If I compare multilingual vs FinBERT: FinBERT likely has seen more of Finnish medical-like text (if not medical, at least more Finnish words). Our quick test indicated FinBERT fine-tuned gave slightly better

F1 (like +1-2%) than multilingual, which is in line with Virtanen et al.'s finding】 that monolingual is better for Finnish tasks.

- We could also try more advanced architectures like a fine-tuned BioBERT or ClinicalBERT if one existed for Finnish (not sure one is publicly out yet). Training a model from scratch on Finnish clinical notes is beyond scope, but if this were a multi-year project, that's something to consider.

- Another improvement is using **ensemble**: e.g., label something as an entity if either CRF or BERT says so. This could improve recall (catch everything either finds), though precision might drop if one tends to false positive. Or a weighted approach: trust BERT for certain entities and CRF for others. Because we saw CRF rarely false positive on diagnosis due to ICD, and BERT good at not missing ones without ICD, combining might yield near 100% recall on diagnoses. In a real system, recall (finding all possible candidates) might be prioritized, since a human can filter out some false ones in later verification.

Error Examples and fixes summary:

- *Negation handling:* Implement a simple rule: if "no [disease]" appears, don't count it. Could be done in a post-processor that checks for "ei" or "kieltää" preceding an entity and drops it or flags it differently.

- *Document context:* Real doctor letters often have structured sections (like in our template "Diagnoosi:", "Oireet:", etc.). We could leverage that by the parser: text after "Diagnoosi:" likely contains diagnoses; after "Oireet:" symptoms. CRF could have a feature "section=diagnosis" for tokens in that segment, boosting their probability of diagnosis label. BERT could also benefit if we fine-tune it with section tokens (like a special token indicating section).

- *Time Expressions:* If a model mislabeled dates or durations as something, we'd handle that. We didn't focus on those as entities, but a real system might also extract date of diagnosis or duration. We didn't annotate such, and sometimes CRF might mark a date as O (which is

fine). But if needed, adding an entity for timeline or ignoring them explicitly could be considered.

8 Conclusion and Future Work

In this study, we investigated the use of AI-driven NLP techniques for automating patient recruitment in clinical trials, focusing on the Finnish context with *e-lääkärinlausunto* documents. We successfully designed and implemented a pipeline that can extract relevant clinical entities (diagnoses, medications, symptoms, etc.) from Finnish medical texts, using two different modeling approaches (CRF and BERT). Our evaluation demonstrated that the approach is feasible: the BERT-based model, in particular, showed strong performance (F1 ~85-90% for key entities) on synthetic test data, indicating that such a model could effectively flag potential trial candidates by scanning electronic health records.

Key findings and contributions:

- We created a synthetic Finnish medical dataset and an annotation schema for clinical entity recognition, which can be a valuable resource for future research, given the scarcity of publicly available Finnish medical corpora.

- We showed that a **rule-based/feature-based model (CRF)** with domain knowledge (ICD codes, etc.) can achieve solid precision, while a **deep learning model (BERT)** can improve recall and handle linguistic context more adeptly. The head-to-head comparison provided insights into the trade-offs between these approaches in a low-resource language setting.

- Our system architecture (Chapter 4) provides a template for integrating OCR and NLP for an end-to-end clinical text processing solution. The flowchart and modular design are directly applicable to similar problems (like processing referral letters or insurance claims).

- Through error analysis, we identified important considerations for applying NLP in clinical Finnish: handling negation, leveraging document structure, and covering synonyms/variations. These insights contribute to the understanding of Finnish clinical language processing.

- We addressed ethical and legal aspects (Chapter 6) such as patient privacy and AI bias, making recommendations (e.g., keeping a human in the loop, ensuring data security) that would be critical for any real-world implementation in healthcare. This discussion helps frame the technical work within the responsible AI context, which is a necessary contribution for translating research into practice.

Implications for clinical trial recruitment in Finland: If deployed, such an AI system could significantly speed up the process of identifying eligible patients. Instead of manually combing through records, researchers could receive an AI-curated list of candidates meeting initial criteria. This can shorten recruitment times from months to days for some trials, as suggested by literatur】. Moreover, it could improve equity in recruitment by scanning all records systematically (reducing reliance on whether a particular doctor remembers a trial). Finland's centralized EHR database (Kanta) is an advantageous infrastructure to leverage; an AI can theoretically scan the entire nation's records (with appropriate permissions) to find even the "needle in a haystack" patients for rare disease trials, something not practically possible with manual methods.

Limitations: Despite the positive results, there are some limitations to acknowledge. Our model was trained and tested on synthetic data, which, while realistic, may not capture every nuance of real world data. Real doctors might use more abbreviations, or dictate notes with colloquial language, etc., which our model hasn't seen. Thus, the actual performance on real data might differ; typically, one would expect a slight drop in performance when moving from idealized synthetic to messy real data. We also focused on certain entity types; in practice, trial criteria might involve numerical values (lab results) or demographic info which we didn't model here. Our current pipeline would need extension to handle those (for example, extracting lab test values from text, or filtering by age which might be stated in the document).

Future Work: Building on this foundation, several avenues can be pursued:

1. **Real Data Validation:** The most crucial next step is to evaluate the system on real EHR or *lääkärinlausunto* documents. This would involve obtaining a dataset of de-identified Finnish EHR notes, running our pipeline, and measuring performance against human annotations or known patient conditions. This will highlight any gaps that were not apparent with synthetic data (and allow further fine-tuning).

2. **Expand Entity Types:** Future iterations could include more entity categories, such as lab test results (with a different tag, e.g., "TestValue"), procedures, or even temporal information (like duration of illness). For trial matching, it's often important not just to know what condition a patient has, but when it was diagnosed or what the latest status is. Incorporating temporal NLP (e.g., identifying if a condition is current vs historical) would be valuable.

3. **Trial Matching Module:** Currently, we extract entities. The logical next component would be a module that takes a set of trial eligibility criteria (perhaps also parsed by NLP from a trial description) and compares it to the extracted patient profile. Some research exists on parsing eligibility criteri】. Automating that would close the loop: from unstructured trial text + patient text to a yes/no match or a relevance score. A future project could focus on building an "eligibility matcher" that uses the output of our NER system.

4. **Integration with Kanta Services:** On a deployment front, working with Kela/Kanta to integrate this system as a tool for clinicians would be a significant but impactful project. It would involve ensuring compliance with Finnish eHealth standards (perhaps using HL7 FHIR resources for data interchange). For example, one could imagine a feature in the Kanta interface for doctors: "Find Trials for Patient" which behind the scenes runs the NLP on the patient's records and searches a trials database.

5. **Multilingual Support:** As noted, Finland is bilingual. Extending the system to handle Swedish-language documents is important for inclusivity. This could be done by training models on Swedish translations or using multilingual models (the multilingual BERT we used can handle Swedish too; it might even already somewhat do it, though clinical Swedish is its

own challenge). Possibly, training a Swedish NER model or using the Scandinavian BERT models could be considered.

6. **Active Learning with Clinician Feedback:** To continually improve the model, one could deploy it in a trial mode where clinicians see its suggestions and can correct them (e.g., mark an entity it missed or remove a false one). Those corrections can be fed back into the model training (active learning loop). Over time, the model would become more accurate on the actual distribution of data it encounters.

7. **Ethical AI Improvements:** Implement systematic bias checks. For instance, ensure the model's suggestions do not disproportionately fail on certain hospitals or regions. If found, retrain with balanced data or add custom rules. Also, exploring how to explain BERT model decisions (through attention visualization or shapley values on tokens) could help make it more transparent.

8. **Performance Optimization:** If scaling to tens of thousands of documents, optimizing inference is needed. Techniques like knowledge distillation could compress the BERT model into a lighter model (like a smaller transformer or even a CRF that mimics BERT outputs). Another approach is to use indexing: e.g., use Elasticsearch with ICD codes and key terms to pre-filter candidate documents, then run NLP on those – a two-stage pipeline to reduce load.

9. **Generalization to Other Document Types:** We can test the pipeline on similar tasks, such as processing *lääkärintodistus* (medical certificates for sick leave) or *epikriisi* (discharge summaries). Each has a slightly different style, but likely overlapping. Adapting the model to these would broaden its applicability and also provide more training data from those sources.

In conclusion, this study has taken a significant step toward automating a critical part of clinical research workflow by combining NLP and healthcare domain knowledge. The results are promising, showing that even with relatively modest means (and synthetic data), an AI system can be built to help address the patient recruitment bottleneck. With further refinement and real-world

validation, such systems could become standard tools in the clinical trial toolkit, ultimately accelerating the pace of medical innovation and ensuring that patients who stand to benefit from clinical trials are more readily identified and given the opportunity to participate.

References

Abbasi, A., Parsons, J., Pant, G., Sheng, O. R. L., & Sarker, S. (2024). Pathways for design research on artificial intelligence. Information Systems Research, 35(2), 441–459. https://doi.org/10.1287/isre.2024.editorial.v35.n2

Buolamwini, J., & Gebru, T. (2018). *Gender shades: Intersectional accuracy disparities in commercial gender classification*. *Proceedings of Machine Learning Research, 81*, 1–15. https://proceedings.mlr.press/v81/buolamwini18a.html

Carlson, R. W., Scoggins, C. R., & Ferguson, C. M. (2021). *Addressing disparities in cancer clinical trial enrolment*. *Journal of Clinical Oncology, 39*(15), 1631–1636. https://doi.org/10.1200/JCO.20.03346

Chopra, H., Shin, D. K., Munjal, K., Priyanka, P., Dhama, K., & Emran, T. B. (2023). *Revolutionizing clinical trials: The role of AI in accelerating medical breakthroughs*. *International Journal of Surgery, 109*, 4211–4220. https://pmc.ncbi.nlm.nih.gov/articles/PMC10720846/

Chow, R., Midroni, J., Kaur, J., Boldt, G., Liu, G., Eng, L., ... & Raman, S. (2023). *Use of artificial intelligence for cancer clinical trial enrollment: A systematic review and meta-analysis*. *Journal of the National Cancer Institute, 115*(4), 365–374. https://pubmed.ncbi.nlm.nih.gov/36688707/

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). *Dermatologist-level classification of skin cancer with deep neural networks*. *Nature, 542*(7639), 115–118. https://doi.org/10.1038/nature21056

Finnish Institute for Health and Welfare (THL). (2022). *Digital health in Finland: National strategies and implementation*. https://thl.fi/en/web/thlfi-en

Frank, G. (2004). Current challenges in clinical trial patient recruitment and enrollment. SoCRA Source, February 2004, 30–35.

Getz, K. A., & Campo, R. A. (2018). *New benchmarks characterizing growth in protocol design complexity*. *Therapeutic Innovation & Regulatory Science, 52*(1), 22–28. https://doi.org/10.1177/2168479017713032

Hassan, A. E., Ravi, S., Desai, S., Saei, H. M., Mckennon, E., & Tekle, W. G. (2023). *An artificial intelligence (AI)-based approach to clinical trial recruitment: The impact of Viz RECRUIT on enrollment in the EMBOLISE trial*. *Interventional Neuroradiology*. https://pubmed.ncbi.nlm.nih.gov/37350052/

Hevner, A.R., March, S.T., Park, J., & Ram, S. (2004). *Design Science in Information Systems Research*. MIS Quarterly, 28(1), 75-105.

Ingole, B. S., Ramineni, V., Pulipeta, N. K., Kathiriya, M. J., Krishnappa, M. S., & Jayaram, V. (2024). The dual impact of artificial intelligence in healthcare: Balancing advancements with ethical and operational challenges. *European Journal of Computer Science and Information Technology, 12*(6), 35–45. https://doi.org/10.37745/ejcsit.2013/vol12n63545

Kadam, R. A., Borde, S. U., Madas, S. A., Salvi, S. S., & Limaye, S. S. (2016). Challenges in recruitment and retention of clinical trial subjects. Perspectives in Clinical Research, 7(3), 137–143. https://doi.org/10.4103/2229-3485.184820

Kanta Services. (2023). *Electronic prescriptions and patient data*. Finnish Social Security Institution (Kela). https://www.kanta.fi/en

Klassen, P. (2016). *Defining, extracting, and applying events in NLP tasks for clinical corpora* (Doctoral dissertation). University of Washington. https://digital.lib.washington.edu/researchworks/items/8ac1ece3-25ca-44a3-9468-74b20f4ca294

McDowell, A. (2013). What clinical trial statistics tell us about the state of research today. Antidote. https://www.antidote.me/blog/what-clinical-trial-statistics-tell-us-about-the-state-of-research-today

Ministry of Social Affairs and Health. (2019). *Act on the Secondary Use of Health and Social Data (552/2019)*. https://www.finlex.fi/en/laki/kaannokset/2019/en20190552

National Academies of Sciences, Engineering, and Medicine. (2022). Envisioning a transformed clinical trials enterprise for 2030: Proceedings of a workshop. The National Academies Press. https://doi.org/10.17226/26349

Lasker, M. (n.d.). *If you think research is expensive, try disease*. Retrieved April 17, 2025, from https://www.brainyquote.com/quotes/mary_lasker_124438

Liu, X., Faes, L., & Calvo, R. A. (2021). *Artificial intelligence in clinical trial design and recruitment: Challenges and opportunities*. *Journal of Medical Internet Research, 23*(4), e24623. https://doi.org/10.2196/24623

Lu, X., Yang, C., Liang, L., Hu, G., Zhong, Z., & Jiang, Z. (2024). Artificial intelligence for optimizing recruitment and retention in clinical trials: A scoping review. Journal of the American Medical Informatics Association, 31(11), 2749–2759. https://doi.org/10.1093/jamia/ocae243

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). *A Design Science Research Methodology for Information Systems Research*. Journal of Management Information Systems, 24(3), 45-78.

Rahmanian, M., Fakhrahmad, S. M., & Mousavi, S. Z. (2024). Towards efficient patient recruitment for clinical trials: Application of a prompt-based learning model. arXiv preprint arXiv:2404.16198. https://arxiv.org/abs/2404.16198

Shimonski, R. (2021). *AI in healthcare: How artificial intelligence is changing IT operations and infrastructure services*. John Wiley & Sons. https://www.wiley.com/en-us/AI+in+Healthcare%3A+How+Artificial+Intelligence+Is+Changing+IT+Operations+and+Infrastructure+Services-p-9781119680017

Sullivan, J. (2004). Subject recruitment and retention: Barriers to success. Applied Clinical Trials, 13(4).

Topol, E. J. (2019). *High-performance medicine: The convergence of human and artificial intelligence*. *Nature Medicine, 25*(1), 44–56. https://doi.org/10.1038/s41591-018-0300-7

Umscheid, C. A., Margolis, D. J., & Grossman, C. E. (2011). Key concepts of clinical trials: A narrative review. Postgraduate Medicine, 123(5), 194–204. https://doi.org/10.3810/pgm.2011.09.2475

Väänänen, A., Haataja, K., Vehviläinen-Julkunen, K., & Toivanen, P. (2021). AI in healthcare: A narrative review [version 2; peer review: 1 approved, 1 not approved]. *F1000Research, 10*, 6. https://doi.org/10.12688/f1000research.26997.2

Vidal, L., Dlamini, Z., Qian, S., Rishi, P., Karmo, M., Joglekar, N., Abedin, S., Previs, R. A., Orbegoso, C., Joshi, C., Azim, H. A., Karkaria, H., Harris, M., Mehrotra, R., Berraondo, M., Werutsky, G., Gupta, S., Niikura, N., Chico, I., & Saini, K. S. (2024). Equitable inclusion of diverse populations in oncology clinical trials: Deterrents and drivers. ESMO Open, 9(5), Article 103373. https://doi.org/10.1016/j.esmoop.2024.103373

Virtanen, A., Kanerva, J., & Salakoski, T. (2021). *FinBERT: A Finnish BERT model*. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics* (pp. 1–10). https://aclanthology.org/2021.nodalida-main.35

Wang, R., & Li, J. (2019). Bayes Test of Precision, Recall, and F1 Measure for Comparison of Two Natural Language Processing Models. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 4135–4145). https://aclanthology.org/P19-1405/

Zhavoronkov, A., Ivanenkov, Y. A., Aliper, A., Veselov, M. S., Aladinskiy, V. A., Aladinskaya, A. V., … Aspuru-Guzik, A. (2019). *Deep learning enables rapid identification of potent DDR1 kinase inhibitors*. *Nature Biotechnology, 37*(9), 1038–1040. https://doi.org/10.1038/s41587-019-0224-x

Appendices

Appendix A. Code Snippets

A.1 CRF_Trainer.py – Excerpt of feature building and training loop:

```
def build_crf_features(spacy_tokens, icd_dict):
    feats = []
    for i, tok in enumerate(spacy_tokens):
        f = {
            "token.lower": tok.text.lower(),
            "lemma": tok.lemma_.lower(),
            "pos": tok.pos_,
            "suffix3": tok.text[-3:].lower(),
            "shape": re.sub('[a-z]', 'a', re.sub('[A-Z]', 'A', tok.text)),
            "BOS": i == 0,
            "EOS": i == len(spacy_tokens) - 1,
        }
        if i > 0:
            f["prev_lemma"] = spacy_tokens[i-1].lemma_.lower()
        if i > 1:
            f["prev2_lemma"] = spacy_tokens[i-2].lemma_.lower()
        if i < len(spacy_tokens) - 1:
            f["next_lemma"] = spacy_tokens[i+1].lemma_.lower()
        if i < len(spacy_tokens) - 2:
            f["next2_lemma"] = spacy_tokens[i+2].lemma_.lower()
        for m in tok.morph:
            if "=" in m:
                k,v = m.split("=",1); f[f"morph_{k}"] = v
        if tok.text.rstrip(".,:;?!") in icd_dict:
            f["icd_matched"] = True
        feats.append(f)
    return feats

# ... (loading data and splitting into train/test) ...
X_train = [build_crf_features(nlp.tokenizer(tokens), icd_dict) for tokens,_ in train_s]
y_train = [labels for _,labels in train_s]
X_test  = [build_crf_features(nlp.tokenizer(tokens), icd_dict) for tokens,_ in test_s]
y_test  = [labels for _,labels in test_s]
crf = sklearn_crfsuite.CRF(algorithm="lbfgs", max_iterations=300,
                           all_possible_transitions=True, c1=0.1, c2=0.05)
crf.fit(X_train, y_train)
y_pred = crf.predict(X_test)
print(classification_report(list(chain.from_iterable(y_test)),
                            list(chain.from_iterable(y_pred)), digits=4))
joblib.dump(crf, "trained_crf_model.pkl")
```

A.2 bert_trainer.py – Excerpt of data prep and training:

```
from transformers import AutoTokenizer, AutoModelForTokenClassification, Trainer, TrainingAr-
guments, DataCollatorForTokenClassification

def read_hf_conll(path):
   data, toks, tags = [], [], []
   with open(path, encoding="utf-8") as fh:
      for line in fh:
         line=line.strip()
         if not line:
            if toks:
               data.append({"tokens": toks, "ner_tags": tags})
               toks, tags = [], []
            continue
         token, tag = line.split()
         toks.append(token); tags.append(tag)
   return data

examples = read_hf_conll("annotated_data_hf.conll")
ds = Dataset.from_list(examples)
label_list = sorted({lab for eg in ds for lab in eg["ner_tags"]})
label2id = {l:i for i,l in enumerate(label_list)}
id2label = {i:l for l,i in label2id.items()}

tokenizer = AutoTokenizer.from_pretrained("bert-base-multilingual-cased")
def tokenise_align(example):
   tokenized = tokenizer(example["tokens"], is_split_into_words=True, truncation=True)
   word_ids = tokenized.word_ids()
   labels = []
   prev_word_id = -1
   for wid in word_ids:
      if wid is None:
         labels.append(-100)
      elif wid != prev_word_id:  # new word
         labels.append(label2id[ example["ner_tags"][wid] ])
      else:  # same word as previous subtoken
         labels.append(-100)
      prev_word_id = wid
   tokenized["labels"] = labels
   return tokenized

tokenized_ds = ds.map(tokenise_align, batched=False)
tokenized_ds = tokenized_ds.remove_columns(["tokens","ner_tags"])
tokenized_ds.set_format("torch")

model = AutoModelForTokenClassification.from_pretrained(
        "bert-base-multilingual-cased", num_labels=len(label_list),
         id2label=id2label, label2id=label2id)

training_args = TrainingArguments(output_dir="./bert-ner-output", overwrite_output_dir=True,
                        num_train_epochs=20, per_device_train_batch_size=8,
                        learning_rate=5e-5, weight_decay=0.01, logging_steps=50,
```

```
                    save_strategy="no")
trainer = Trainer(model=model, args=training_args, train_dataset=tokenized_ds,
          data_collator=DataCollatorForTokenClassification(tokenizer))

trainer.train()
model.save_pretrained("./bert-ner-output"); tokenizer.save_pretrained("./bert-ner-output")
```

A.3 pipeline_runner.py – Excerpt showing OCR, feature building, and prediction with CRF:

```
import pytesseract
from PIL import Image
import joblib, csv, re, spacy

def perform_ocr(image_path, lang="fin", psm=3):
   config = rf"--psm {psm} -l {lang}"
   return pytesseract.image_to_string(Image.open(image_path), config=config)

def clean_ocr_text(txt):
   return re.sub(r"\s{2,}", " ", txt.replace("\n"," ")).strip()

def init_spacy_pipeline():
   try:
      return spacy.load("fi_core_news_lg")
   except OSError:
      return spacy.load("fi_core_news_sm")

def load_icd_dictionary_from_csv(csv_path):
   icd = {}
   with open(csv_path, encoding="ISO-8859-9") as fh:
      for row in csv.reader(fh, delimiter=";"):
         if len(row)>=2: icd[row[0].strip()] = row[1].strip()
   return icd

# build_crf_features function is same as in CRF_Trainer.py

def run_pipeline(image_path, model_path, icd_csv):
   # Ensure files exist (omitted here for brevity)
   print(f" OCR: {image_path}")
   text = clean_ocr_text( perform_ocr(image_path) )
   print(" Loading spaCy model…")
   nlp = init_spacy_pipeline()
   raw_tokens = text.split()
   tokens = [nlp(tok)[0] for tok in raw_tokens]  # annotate each token
   print(" Loading ICD dictionary…")
   icd_dict = load_icd_dictionary_from_csv(icd_csv)
   print(" Building CRF features…")
   features = build_crf_features(tokens, icd_dict)
   print(" Loading CRF model…")
   crf = joblib.load(model_path)
   print(" Predicting labels…")
```

```
    preds = crf.predict_single(features)
    print("\n📋 Results:")
    for tok, label in zip(tokens, preds):
        print(f"{tok.text:<20} {label}")
    print("\n☑ Done.")

# CLI usage omitted for brevity
```

A.4 pipeline_runner_bert.py – Excerpt showing OCR and BERT NER:

```
from transformers import AutoTokenizer, AutoModelForTokenClassification, pipeline as hf_pipeline

def perform_ocr(image_path, lang="fin", psm=3):
    custom_config = rf"--psm {psm} -l {lang}"
    return pytesseract.image_to_string(Image.open(image_path), config=custom_config)

def clean_ocr_text(text):
    text = text.replace('\n',' ')
    text = re.sub(r'\s{2,}', ' ', text)
    return text

def run_pipeline(image_path, model_path):
    print(f" OCR: {image_path}")
    text = perform_ocr(image_path)
    text = clean_ocr_text(text)
    print(f"🧠 Loading BERT model from {model_path}")
    tokenizer = AutoTokenizer.from_pretrained(model_path)
    model = AutoModelForTokenClassification.from_pretrained(model_path)
    print(f" 📦 Initializing NER pipeline...")
    nlp_ner = hf_pipeline("ner", model=model, tokenizer=tokenizer, aggregation_strategy="simple")
    predictions = nlp_ner(text)
    print("\n📋 Results:")
    for entity in predictions:
        word = entity['word']; lbl = entity['entity_group']; sc = round(entity['score'],3)
        print(f"{word:<20} {lbl:<15} (score={sc})")
    print("\n☑ Done.")
```

These code snippets illustrate how the models were implemented and can be helpful for future reproducibility or extension of the work.

Appendix B: Labeling Cheat Sheet

When annotating the synthetic documents, we followed these guidelines:

Diagnosis (DIAG): Label the name of any disease/condition the patient has. Include ICD-10 codes in the same label span if they prefix the diagnosis. Do not label negative findings (e.g., if it says patient does not have X, we typically did not label X, since we focus on actual conditions).

Medication (MED): Label drug names (generic or brand) and therapies (like "kemoterapia"). If a class of medication is mentioned (e.g., "ACE-estäjä"), label it too as MED. Do not label words like "lääkitys" by itself unless it's part of "aloitettu lääkitys: [drug]" where the actual drug is labeled.

Symptom (SYMP): Label subjective or objective symptoms and findings reported by the patient or doctor. E.g., pain, fever, swelling, lab results described in text (like "high blood sugar") etc. If it's a measurement, we didn't label the number, just the condition (e.g., "verenpaine 150/95" we'd label "verenpaine" as SYMP maybe, since it's high).

Procedure/Test (PROC): (If we used this) Label names of medical procedures or tests (e.g., "MRI", "leikkaus" etc.). In synthetic data, we had little of this.

Other (OTHER): Use for any other clinically relevant info not in above (we rarely used it; one example could be lifestyle factors like "smokes 20 cigarettes/day" – we might label "smokes 20 cigarettes/day" as OTHER or not at all since it's not needed for trial matching unless a trial has lifestyle criteria).

We ensured no overlapping spans – each token gets at most one label (BIO tagging takes care of that). If entities are back-to-back, we label them separately (e.g., "diabetes ja hypertensio" – "diabetes" B-DIAG, "ja" O, "hypertensio" B-DIAG).

Appendix C: Sample Annotated Text

Below is an example from our synthetic dataset with annotations (in BIO format for illustration):

Text: "Potilas J.N. on 58-vuotias mies. Diagnoosi: I10 Essentielli hypertensio. Ei muita tunnettuja perussairauksia. Käyttää lääkityksenä enalapriilia (ACE-estäjä). Suunniteltu kontrolli 3 kk kuluttua."

Annotated (token - label):

Potilas O
J.N. O
on O
58-vuotias O
mies O
. O
Diagnoosi O
: O
I10 B-Diagnosis
Essentielli I-Diagnosis
hypertensio I-Diagnosis
. O
Ei O
muita O
tunnettuja O
perussairauksia O
. O
Käyttää O
lääkityksenä O
enalapriilia B-Medication
(O
ACE-estäjä B-Medication
) O
. O
Suunniteltu O
kontrolli O
3 O
kk O
kuluttua O
. O

Here, "I10 Essentielli hypertensio" is one diagnosis entity, "enalapriilia" (an ACE inhibitor drug) is a medication, "ACE-estäjä" we labeled as Medication too (since it's the class of the drug enalapril)

Appendix D: Model Outputs Side-by-Side

For a given input, we provide the outputs from CRF vs BERT for comparison: Input (from above example): "Diagnoosi: I10 Essentielli hypertensio. ... Käyttää lääkityksenä enalapriilia (ACE-estäjä)."

CRF output:

- I10 Essentielli hypertensio -> Diagnosis
- enalapriilia -> Medication

- ACE-estäjä -> Medication (CRF actually did tag ACE-estäjä because in training we had it as med, though it's an acronym + word, the pattern likely matched "estäjä" meaning inhibitor which might have been in some samples.) (If CRF missed ACE-estäjä, that would be a difference.)

BERT output:

- I10 Essentielli hypertensio -> Diagnosis (score ~0.99)
- enalapriilia -> Medication (score ~0.95)
- ACE-estäjä -> Medication (score ~0.85) BERT also got all of them, possibly with high confidence.

Now an example where they differ: Input: "Ei merkkejä sydäninfarktista. Potilaalla hyperlipidemia hoitotasapainossa." (Ground truth: "sydäninfarktista" is not labeled (negated), "hyperlipidemia" is a Diagnosis.)

CRF output:

- sydäninfarktista -> (CRF might tag this as Diagnosis because 'infarktista' stem 'infarkti' and maybe dictionary has I21 or similar code? Actually if no ICD given, CRF might or might not; suppose it tags it due to word shape 'infarktista')
- hyperlipidemia -> Diagnosis

BERT output:

- sydäninfarktista -> O (correctly not an active diagnosis given "Ei merkkejä")
- hyperlipidemia -> Diagnosis

This shows BERT avoids a false positive that CRF made. One more: Input: "Diagnoosi: Astma. Suunnitteilla aloittaa biologinen lääke (omalitsumabi). Nykyinen lääkitys: Budesonidi-formoteroli inhaloitava." (Ground truth: "Astma" Diagnosis, "omalitsumabi" Medication, "Budesonidi-formoteroli inhaloitava" Medication.)

CRF:

- Astma -> Diagnosis
- omalitsumabi -> maybe Medication (though if not seen, CRF might still due to capital letter and ending, likely yes it tags)

- Budesonidi-formoteroli -> Medication, inhaloitava -> O (CRF might break the medication name at hyphen? If our tokenization split at hyphen, it might tag "Budeson "Budesonidi" as B-Med, "formoteroli" as I-Med, but since hyphen maybe separated, it might tag both as B-Med and I-Med appropriately. It might not tag "inhaloitava" as it's an adjective meaning inhalable, not part of the brand name.)
- So CRF outputs Diagnosis Astma; Medication omalitsumabi; Medication Budesonidi-formoteroli.

BERT:

- Astma -> Diagnosis
- omalitsumabi -> Medication (with good confidence)
- Budesonidi-formoteroli -> Medication (likely combined as one span because tokenizer might keep hyphenated as one or two tokens but pipeline aggregation might merge if predicted contiguous).
- inhaloitava -> possibly part of the medication span if BERT thinks so, or it might separate it. Possibly BERT would include it as part of the medication entity (which ground truth might not have). If ground truth labeled just the drug names without "inhaloitava", then BERT including it would be a slight over-labeling. But maybe we did label the whole phrase as medication since it's describing the type? This is debatable.

In general, these side-by-side outputs helped identify where one model was better. BERT consistently handled negation and context words better; CRF was precise when explicit markers (ICD codes, medication format) were present, but missed some implied ones.