

## ORIGINAL RESEARCH ARTICLE

From raw text to cross-framework training data:  
Building medical named entity recognition  
datasets with FIT4NERFlorian Freund<sup>\*</sup>, Philippe Tamla<sup>1</sup>, Sven Stieber<sup>1</sup>, and Matthias Hemmje<sup>1</sup>Chair of Multimedia and Internet Applications, Faculty of Mathematics and Computer Science,  
University of Hagen, Hagen, North Rhine-Westphalia, Germany

## Abstract

High-quality annotated medical text data are essential for training robust machine learning-based named entity recognition (NER) models, particularly for extracting structured evidence from large volumes of unstructured medical literature to support the development of clinical practice guidelines. This article introduces a system for collecting, annotating, and managing high-quality training data for machine learning-based NER models. The system is designed to help medical professionals create and maintain extensive training and test datasets across multiple text formats and for different NER frameworks. It also supports the straightforward integration of new NER frameworks through customizable converters. Using the Nunamaker methodology for a structured approach to information system development, the article starts with an introduction to the topic, contextualizes the research, reviews the state of the art, and identifies challenges in text annotation by medical experts. This is followed by a description of the system's modeling and implementation. The article concludes with an expert evaluation of the system, the resulting insights, and a summary of the main findings.

**Keywords:** Named entity recognition; Machine learning; Clinical practice guidelines; Information extraction; Cloud; Data preprocessing

**\*Corresponding author:**Florian Freund  
(florian.freund@fernuni-hagen.de)

**Citation:** Freund F, Tamla P, Stieber S, Hemmje M. From raw text to cross-framework training data: Building medical named entity recognition datasets with FIT4NER. *J Clin Inform.* 2026;2(1):025420035. doi: 10.36922/JCI025420035

**Received:** October 14, 2025**Revised:** January 30, 2026**Accepted:** March 17, 2026**Published online:** May 15, 2026

**Copyright:** © 2026 Author(s). This is an Open-Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

**Publisher's Note:** AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## 1. Introduction and motivation

High-quality annotated data are crucial for developing powerful machine learning (ML) models for named entity recognition (NER) that can support the development of clinical practice guidelines (CPGs) in the medical field.<sup>1</sup> CPGs are evidence-based recommendations that help physicians make decisions about the optimal treatment of their patients.<sup>1</sup> Employing CPGs can subsequently help minimize patient risk during clinical decision-making.<sup>2</sup> Natural-language medical texts, including clinical trial reports, case studies, and specialized literature, contain large amounts of valuable information, which can be utilized for developing CPGs.<sup>1</sup> The volume of medical documents can be extremely large, resulting in information overload and making manual analysis challenging.<sup>3</sup> Domain experts often face the challenge of extracting relevant information for CPG development due to the vast amount of unstructured data.<sup>4</sup> By transforming unstructured text into structured data, ML-based NER aids in managing information overload within the medical field.<sup>1</sup> These structured data can support information

retrieval processes for evidence retrieval, forming the foundation of CPGs.<sup>1</sup> Cloud computing can enhance ML training by providing scalable resources, enabling faster processing and analysis of large datasets.<sup>5</sup> To develop robust ML models for NER, it is imperative to optimize the quantity and quality of training and test data. As stated in Konkol<sup>6</sup>(p22), “a complex system trained on too few data will not be able to generalize, and results on unseen data can be catastrophic.” Generalization refers to the model’s ability to accurately perform on new data points that were not part of the training set.<sup>6</sup>

Building on this need for annotated data in CPG development, the present research is motivated by the following research projects. RecomRatio, launched in 2018, aims to assist medical professionals in making therapy decisions.<sup>7</sup> It uses the spaCy framework and ML-based NER to extract new knowledge from the medical literature.<sup>8</sup> The Artificial Intelligence for Hospitals, Healthcare & Humanity (AI4H3) project, building on the results of RecomRatio, focuses on improving the transparency and explainability of medical decisions.<sup>9</sup> It incorporates artificial intelligence (AI) and a hub architecture to efficiently integrate additional information sources, e.g., medical and biological literature, electronic health records, and healthcare-related social media data, crucial for clinical decision support.<sup>10</sup> The project leverages a knowledge management system called Content and Knowledge Management Ecosystem Portal (KM-EP), developed jointly by the Faculty of Mathematics and Computer Science of the University of Hagen, Chair of Multimedia and Internet Applications<sup>11</sup>, and FTK e.V. Research Institute for Telecommunications and Cooperation.<sup>12</sup> KM-EP facilitates knowledge management in multiple domains<sup>13</sup> and has been used in various research projects, including the H2020 projects Metaplat<sup>13</sup>, SenseCare<sup>14</sup>, and RecomRatio.<sup>7</sup> Additionally, the Stanford Named Entity Recognition and Classification (SNERC)<sup>15</sup> project enables customized NER models for applied games<sup>16–18</sup>, although with limited support for domain expert annotation, using Stanford CoreNLP.<sup>15</sup> By contrast, the Cloud-based Information Extraction (CIE) project aims to enable end-to-end NER pipelines in a cloud environment, optimizing resource allocation.<sup>5</sup> The Framework-Independent Toolkit for Named Entity Recognition (FIT4NER) project, embedded within the AI4H3 and CIE contexts,<sup>19</sup> uses KM-EP to support medical experts in employing various AI-based text analysis techniques such as NER, event identification, and relation extraction<sup>10</sup>, for efficient information retrieval in the development of CPGs. In KM-EP, documents such as medical research findings and clinical studies are managed and used as an evidence base for developing CPGs.<sup>1,19</sup> To optimize information retrieval, KM-EP offers document classification features

using NER, along with a faceted search engine based on these classifications.<sup>15</sup> The necessary ML-based NER models are integrated into KM-EP and should be trained directly by experts in the medical domain.<sup>1</sup> To effectively support medical professionals in developing CPGs through FIT4NER and KM-EP, it is crucial to use powerful ML models for NER.<sup>1</sup> The performance of such models is highly dependent on the size and quality of the training data in the relevant target language<sup>20</sup>, derived from the corresponding application domain.<sup>19</sup> The seamless integration of annotation functions in KM-EP supports the preprocessing of training data for NER.

In the medical field, highly specialized domains such as biomedicine and clinical practice rely on a multitude of domain-specific terms and abbreviations for medical procedures, tests, conditions, and treatments.<sup>21,22</sup> However, these terms and abbreviations can often be ambiguous or exhibit variations in spelling, which poses challenges for extraction using ML methods. Therefore, involving medical experts in creating annotations that account for domain-specific terms, abbreviations, and potential misspellings is crucial for training new models and achieving optimal results.<sup>23</sup> Nevertheless, the performance and suitability of NER frameworks vary depending on the use case<sup>24</sup>, necessitating the comparison and selection of a suitable framework for each specific application.<sup>19</sup> Moreover, different NER frameworks require different text and annotation formats, such as CoNLL-2003<sup>25</sup>, BRAT<sup>26</sup>, or BioC.<sup>27</sup> In this work, these formats serve different roles in the preprocessing pipeline: BioC is used as the internal representation to harmonize heterogeneous source documents and retain expressive annotation structures, while BRAT and CoNLL-2003 are supported at the interfaces to maximize interoperability with existing corpora and widely used NER frameworks. CoNLL-2003<sup>25</sup> is a line-based sequence-labeling format that is broadly supported by NER toolkits, but it cannot represent discontinuous spans, nested entities, or relations. BRAT<sup>26</sup> is a widely adopted annotation environment that supports richer annotation structures (e.g., relations and discontinuous annotations) and is commonly used for biomedical corpora. BioC<sup>27</sup> is an XML-based interchange format designed for biomedical texts and annotations, providing a structured representation that supports interoperability across tools. Because NER methods and toolchains evolve quickly, future frameworks may introduce new data requirements; therefore, the system must support multiple formats and allow new converters to be integrated with limited engineering effort.<sup>19</sup>

Having highlighted the critical importance of helping medical experts in annotating domain-specific texts for

NER model training, the research questions (RQs) for this work are defined below. Medical professionals face the challenge of finding suitable tools to create and manage training data in various formats, which also support future text and annotation formats. This issue leads to two RQs examined in a bachelor's thesis based on FIT4NER<sup>28</sup>:

- (i) How can a system be developed and integrated into KM-EP to support medical experts in the creation of high-quality training and test data for ML-based NER?
- (ii) How can training and test data be used efficiently in a cloud environment to train ML models with various NER frameworks?

The goal of this research is to support medical experts through a flexible, cloud-based system integrated into KM-EP that facilitates the creation of high-quality data for ML-based NER models and efficiently processes large datasets.

This article employs the Nunamaker methodology<sup>29</sup>, a systematic approach for developing information systems, which includes several research objectives (ROs) spanning observation, theory building, implementation, and evaluation phases.

The remainder of the article is structured as follows: Section 2 addresses the observation objectives, reviewing the current state of the art. Section 3 focuses on theory-building objectives, developing models to aid medical experts in creating training and test data. Section 4 outlines the system development objectives, describing the creation of a prototype system for generating training and test data. Section 5 covers the experimentation objectives, describing the execution and analysis of expert tests using the prototype. Finally, Section 6 summarizes the results of the study.

## 2. State of the art in science and technology

This section focuses on the observation phase and RO1, which entails the “review of the current state of the art.” The objective is to identify and discuss remaining challenges (RCs) in the areas addressed in this article.

Within the SNERC project<sup>15</sup>, use cases such as “Annotate Domain Corpus,” “Select Data Cleanup Options,” “Generate/Upload Training Data,” and “Generate/Upload Testing Data” were defined.<sup>15</sup> These use cases were subsequently extended in the CIE project to include functionality supporting the training of NER models using cloud resources.<sup>5</sup> FIT4NER, an extension of CIE, further emphasizes the comparison and selection of different NER frameworks.<sup>19</sup> As a result, the SNERC use cases were consolidated into the “Support Data Preparation” use case.<sup>19</sup>

Building on these efforts, the “Process Model for AI-based Knowledge Extraction Support for CPG development”<sup>19</sup> was derived, illustrating the essential process steps to train NER models in support of the CPG development. This model explicitly includes the “Data Management & Curation” phase, which encompasses all activities related to data preparation. Efficient data preparation and utilization are fundamental prerequisites for effective ML-based NER model training.<sup>30</sup> Consequently, medical professionals must be equipped with robust tools that support the creation, management, and annotation of training data in multiple formats.

When creating high-quality datasets for NER, several challenges arise as early as the conceptual stage. First, a precise annotation scheme is required, defining a clear taxonomy of entity types, unambiguous entity boundaries, and a consistent differentiation between closely related classes. For example, the Colorado Richly Annotated Full Text (CRAFT) corpus relies on multiple ontologies such as ChEBI<sup>31</sup> and Uberon<sup>32</sup>, whereas the BRONCO corpus assigns drugs according to the Anatomical Therapeutic Chemical Classification System.<sup>33</sup> In addition, the selection of texts to be annotated must be systematic to ensure representative coverage of all relevant entity types. In highly class-imbalanced domains, stratified sampling is often necessary to adequately represent rare entities.<sup>34</sup> Domain-specific characteristics further complicate the annotation process. Medical texts frequently contain lexical ambiguities (e.g., cold as a disease or a temperature)<sup>35</sup>, discontinuous or nested entity structures<sup>36</sup>, and context-dependent phenomena such as negations (“no cold present”) or temporal references (“previously existed ...”).<sup>33</sup> Moreover, unequal entity frequencies lead to discrepancies in recognition performance: while frequent terms (e.g., fever, pain) are typically recognized robustly, rare symptoms or therapies are identified with significantly lower accuracy.<sup>34</sup> External constraints, such as the General Data Protection Regulation (GDPR), may further degrade corpus quality when anonymization procedures remove clinically relevant contextual information.<sup>37</sup> Finally, the identification of newly emerging entities and the limited availability of domain experts—whose time is limited—pose additional challenges.<sup>8</sup> To address these issues, a multi-stage annotation strategy is commonly recommended. In an initial pilot phase, the annotation schema and associated guidelines are tested on a small subset of documents to identify ambiguities and refine definitions before large-scale annotation begins.<sup>38</sup> The practical annotation process is typically supported by specialized tools such as BRAT, which was used in the studies by Cohen *et al.*,<sup>31</sup> Kittner *et al.*,<sup>33</sup> and Nastou *et al.*<sup>38</sup> and provides intuitive user interfaces and consistency checks. Reducing the workload for domain

experts is a central objective and can be achieved through hybrid annotation approaches, in which automated pre-annotations are generated and subsequently validated and refined by experts.<sup>33</sup> Recent approaches, such as NERFlow, further integrate large language models into configurable annotation pipelines to support this process.<sup>39</sup> Additionally, active learning techniques enable the iterative selection of highly informative samples—often characterized by high model uncertainty—thereby improving model quality while minimizing annotation effort.<sup>40</sup> The consistency and reliability of manual annotation are further promoted through structured annotator training, detailed guideline documents, and the use of exemplary annotation schemes. Employing multiple annotators for the same data increases reliability, while remaining disagreements are resolved through systematic adjudication by experienced annotators or expert panels.<sup>31,33</sup> For quality assurance, quantitative measures of inter-annotator agreement (IAA) are commonly used to assess annotator consistency and to evaluate the clarity and effectiveness of the annotation scheme.<sup>41</sup> Together, these measures enable the efficient creation of consistent and representative NER corpora, even in complex and highly domain-specific application areas. In practice, these quality factors translate directly into measurable differences in model performance: ambiguous boundaries, inconsistent guidelines, and unresolved annotator disagreement typically reduce precision and recall, while pilot phases, adjudication, and IAA-driven guideline refinement improve downstream NER robustness.<sup>31,33,41</sup>

To ensure transparency for both medical and technical audiences, the selection of annotation formats and associated tools in this work was guided by several criteria, including expressiveness, interoperability, standardization and structure, community adoption and maintenance, and practical feasibility within expert workflows. Specifically, these criteria included: (i) expressiveness, including support for discontinuous spans, nested entities, and relations; (ii) interoperability, reflected by the availability of converters and compatibility with existing NER frameworks and corpora; (iii) standardization and structure, such as the availability of formal schemas (e.g., XML-based formats); (iv) community adoption and maintenance, indicating an active ecosystem and sustained tool support; and (v) practical feasibility within expert workflows, including the availability of usable annotation environments and a manageable engineering effort for integration. These criteria reflect the dual objective of this work—to represent biomedical annotations with sufficient fidelity during preprocessing while enabling seamless export to widely used NER training pipelines. Medical texts exist in varied formats such as PDF, DOCX,

HTML, or plain text.<sup>42,43</sup> To support medical experts in annotating and reusing these texts across different NER frameworks, the preprocessing pipeline must convert heterogeneous source documents into a single, stable internal representation and then provide framework-specific exports.<sup>27</sup> We therefore compared annotation formats with respect to expressiveness (e.g., discontinuous, overlapping, or nested annotations), structural properties (e.g., XML schema), and ecosystem support (availability of tools and converters).<sup>27,44</sup> The evaluated formats included CoNLL-2003<sup>25</sup>, GPML<sup>45</sup>, BRAT<sup>26</sup>, Knowtator<sup>46</sup>, BioC<sup>27</sup>, and CLAO.<sup>47</sup> BioC was selected as the internal representation because it provides a structured, XML-based model and is widely used for biomedical interoperability.<sup>27</sup> In contrast, CoNLL-2003 was retained as a supported export format despite its limited expressiveness (no discontinuous spans, relations, or nested entities), because it remains a *de facto* standard for many sequence-labeling training pipelines and is supported by numerous NER frameworks.<sup>25</sup> BRAT was treated as a tool-facing interface format for importing or editing corpora, because it is frequently used in biomedical annotation practice and supports richer annotation structures through its standoff representation.<sup>26</sup> GPML cannot represent discontinuous annotations, relations, or nested named entities<sup>45</sup>, and CLAO and Knowtator were excluded in this work due to limited practical adoption and tooling support in the targeted workflow.<sup>31,47</sup>

Table 1 summarizes the key feature differences across the analyzed formats and should be interpreted as a capability overview. Within the proposed architecture, expressive formats such as BioC and BRAT can be preserved internally or during expert annotation, whereas simpler formats such as CoNLL-2003 are provided as export formats to ensure compatibility with common training pipelines. Given the continuous emergence of new document types and NER toolchains, Prepare4NER adopts a modular converter architecture that allows format-specific converters to be added or replaced without impacting the remainder of the system.<sup>47</sup> Leveraging different ML models and NER frameworks for model training requires efficient data preparation and use.<sup>19</sup> Cloud computing offers scalable resources for managing the substantial storage and computational demands of NER model training.<sup>5,48</sup> The use of cloud computing technologies for sensitive medical data requires globally coordinated measures to comply with regulatory requirements. In the European Union, for example, the GDPR provides a stringent legal framework that must be adhered to, particularly when processing personal data.<sup>49</sup> Selecting an appropriate cloud provider is crucial, ensuring data storage complies with legal requirements and that compliance is maintained through contractual agreements, such as data processing

**Table 1. Annotation formats and supported features (capabilities and ecosystem factors used for format selection)**

Name	XML	DA	Nested NEs	Relations	Active community
CoNLL-2003 <sup>25</sup>	–	–	–	–	X
GPML <sup>45</sup>	X	–	–	–	–
BRAT <sup>26</sup>	–	X	X	X	X
Knowtator <sup>46</sup>	X	X	X	X	–
BioC <sup>27</sup>	X	X	X	X	X
CLAO <sup>47</sup>	X	X	X	X	–

Notes: “X” indicates that the format supports the feature or satisfies the criterion; “–” indicates that the feature is not supported, not applicable, or not sufficiently available.

Abbreviations: DA: Discontinuous annotation; NE: Named entity.

agreements. Technical measures such as anonymization, pseudonymization, and end-to-end encryption are essential to ensure data security and protection.<sup>49</sup> Specialized medical clouds and international certifications, such as ISO 27001, further support compliance with these regulations.<sup>49,50</sup> Additionally, technologies such as Kubernetes enable independence from specific providers, facilitating flexible processing of medical data in public clouds, specialized clouds, or on-premise private clouds.<sup>50,51</sup> These cloud orchestration tools have revolutionized container management and deployment, providing scalable and efficient deployment strategies.<sup>50,51</sup> The ability to perform autoscaling is particularly critical as it allows resources to be dynamically adjusted according to demand, which is essential for handling large datasets.<sup>51</sup> The measures can help mitigate, although not eliminate, the regulatory and technical challenges of using cloud computing in the medical context. In the context of CIE, major cloud providers such as Amazon Web Services and Microsoft Azure have already been used to support the training of NER models with large datasets.<sup>5</sup> Consequently, it should be feasible to provide annotated training and test data on the cloud storage systems of these leading providers.

The knowledge management system KM-EP is widely adopted in research<sup>7,13,14</sup> and is therefore well suited to manage medical information that can be used in the development of CPGs.<sup>1</sup> In addition, it provides a unified user interface and supports cross-functional features such as user and access-rights management.<sup>13</sup> For these reasons, KM-EP was selected as the foundation for the FIT4NER project.<sup>19</sup> This work is embedded within FIT4NER and therefore requires support for annotating medical texts within KM-EP and for the subsequent use of these annotations to train ML models for NER.

The authors identified two RCs essential for developing an effective system architecture. The first RC involves enabling flexible integration of document conversion tools into KM-EP to support specific formats efficiently and

seamlessly integrate different document types into the annotation workflow. The BioC format was selected for internal representation because it is the most expressive among the formats studied. The second RC focuses on facilitating the provisioning of training and test data on the storage resources of the major cloud providers, to support training with NER frameworks that utilize the selected data formats. This requirement highlights the importance of optimizing data accessibility and utilization in multiple environments to enhance model training and deployment capabilities.<sup>5</sup> These requirements underscore the importance of designing a versatile system architecture capable of supporting various formats and environments to optimize ML-based NER model training and deployment.

### 3. Conceptual modeling and design

This section focuses on the theory-building phase, addressing RO2, “Creation of essential models for a system that allows the creation of training and test data by medical experts.” This is achieved by considering the RCs presented in Section 2, based on the current state of the art in science and technology. User-centered system design<sup>52</sup> was applied for design and conceptual modeling, and Unified Modeling Language<sup>53</sup> was used as the specification language. To facilitate the development of ML models for NER in support of CPG development, the “Process model for AI-based knowledge extraction support for CPG development” was initially conceptualized and published.<sup>1(p7)</sup> This model facilitates the comparison and selection of NER frameworks to build domain-specific models for NER. The annotation of unstructured medical texts is identified as a fundamental step in the “Data Management & Curation” phase within this process model. As part of the FIT4NER project<sup>19</sup>, use cases and a potential architecture for a system that implements all steps of this process model were developed. One crucial component of this architecture is the “Model Definition Manager”<sup>19</sup>, which leverages the “NER Framework Independent Service”<sup>19</sup> to preprocess data for training ML models in NER. This section develops

The use cases proposed in this work address specific features of both FIT4NER<sup>19</sup> and CIE<sup>5</sup>, as shown in [Figure 1](#). The use cases are broadly classified into two main topics: “Data Preparation” and “Data Provision.” Under “Data Preparation,” the use cases include “UC1.1 Visualize Data,” “UC1.2 Clean Data,” “UC1.3 Annotate Data,” and “UC1.4 Split Dataset.” “Data Provision”

The diagram illustrates the FIT4NER system architecture and its use cases. It is organized into three main sections: Actors, FIT4NER Use Cases, and Prepare4NER Use Cases.

**Actors:**

- Domain Expert:** Represented by a stick figure icon.
- Administrator:** Represented by a stick figure icon.

**FIT4NER Use Cases (Green Ovals):**

- UC1 Compare NER Frameworks:** Includes UC1.1 Support Data Preparation, UC1.2 Support Model Training, and UC1.3 Evaluate and validate Models.
- UC1.1 Support Data Preparation:** Includes UC1.1.1 Visualize Data, UC1.1.2 Clean Data, UC1.1.3 Annotate Data, and UC1.1.4 Split Data.
- UC1.2 Support Model Training:** Includes UC1.2.1 Store Model and UC1.2.1.1 Store in Cloud.
- UC2 Add NER Framework:** Includes UC2.1 Add Support for Document Format, UC1.2.1 Load Training Data, UC1.3.1 Load Test Data, and UC2.2 Add Support for Annotation Format.

**Prepare4NER Use Cases (Grey Ovals):**

- Data Preparation:** Includes UC1.1.1 Visualize Data, UC1.1.2 Clean Data, UC1.1.3 Annotate Data, UC1.1.4 Split Data, and UC2.1 Add Support for Document Format.
- Data Provision:** Includes UC1.2.1 Load Training Data, UC1.3.1 Load Test Data, and UC2.2 Add Support for Annotation Format.

**Legend:**

- Use-Case:** Represented by a grey oval.
- Prepare4NER Use-Cases:** Represented by a green oval.
- FIT4NER Use-Cases:** Represented by a blue oval.
- CIE Use-Cases:** Represented by a light blue oval.

doi: 10.36922/JCI025420035



The internal representation of the documents uses the structuring elements defined by the BioC format<sup>27</sup>, which are collection, document, passage, and sentence. A document element represents a single document converted from an external data format, which can contain several annotation elements. Documents are always part of a BioC collection, which thus represents a set of documents. BioC also defines an infon element that can be used to provide additional information on each level of the BioC structure. In this work, the infon element is used on the dataset and document levels to add information such as IDs and captions, as well as information on whether a document is used for training or testing. Listing 1 presents a BioC key file that describes these types of information, which are mandatory for each collection and document. The key file is also part of the BioC specification and can be referenced in the *collection* element of a BioC document.<sup>27</sup> Other information models are the document formats of the original documents, as well as the format of pre-annotated corpora on the one hand, and the annotation formats used by the NER frameworks that are used for model training on the other hand. Between these data models, data conversion steps must be integrated into the system.

```
collection:
  infon datasetId:
    unique id to
    identify this
    dataset infon
  datasetName:
    descriptive name of
    this dataset
  document: infon
    title: descriptive
    or original title of
    the document infon
    usage:
      UNDEFINED usage of this document
      is not defined/dataset is not
      split
      TRAINING document
      is to be used for
      model training
      TEST document is
      to be used for
      model evaluation
```

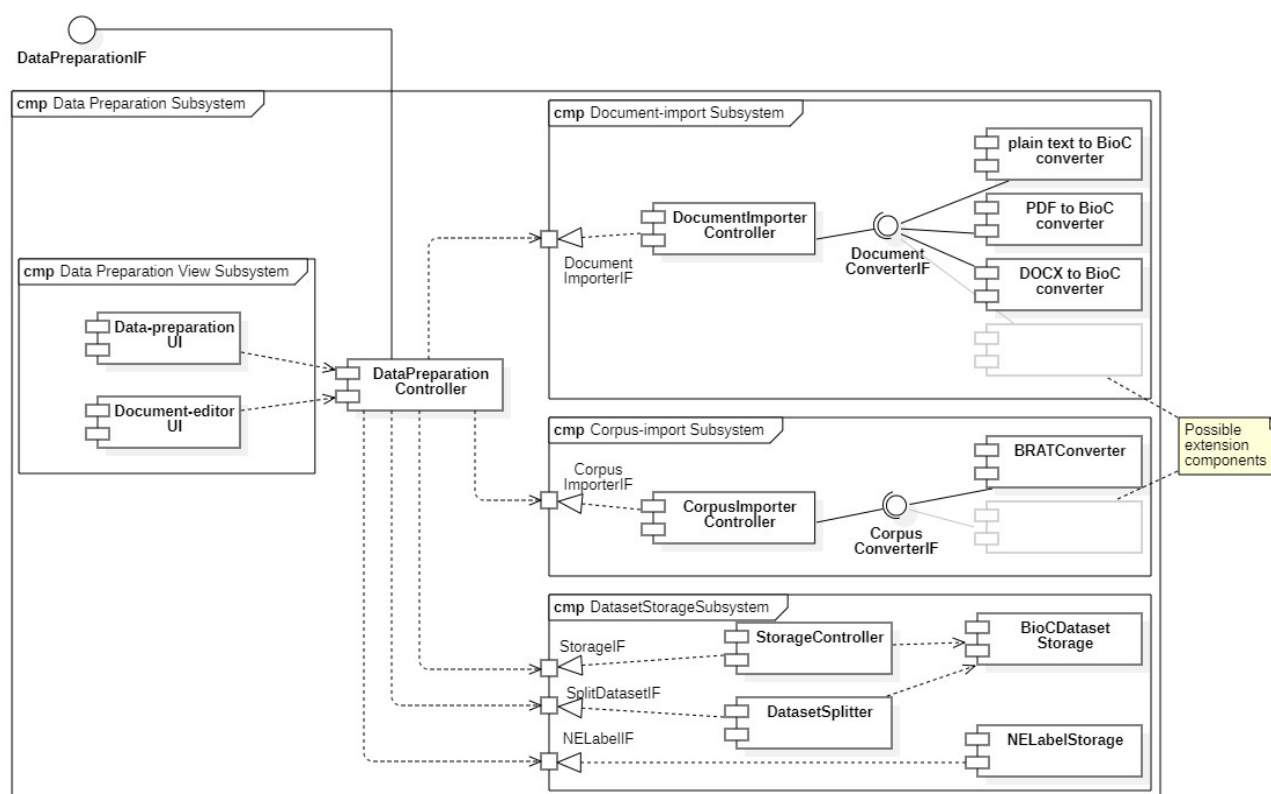
**Listing 1.** BioC key file with definition of mandatory infon types.

To achieve the necessary flexibility for integrating new NER frameworks and document formats, as outlined in the RCs, the “Strategy Pattern”<sup>54</sup>(p373) was used. In implementation, these algorithms, referred to as

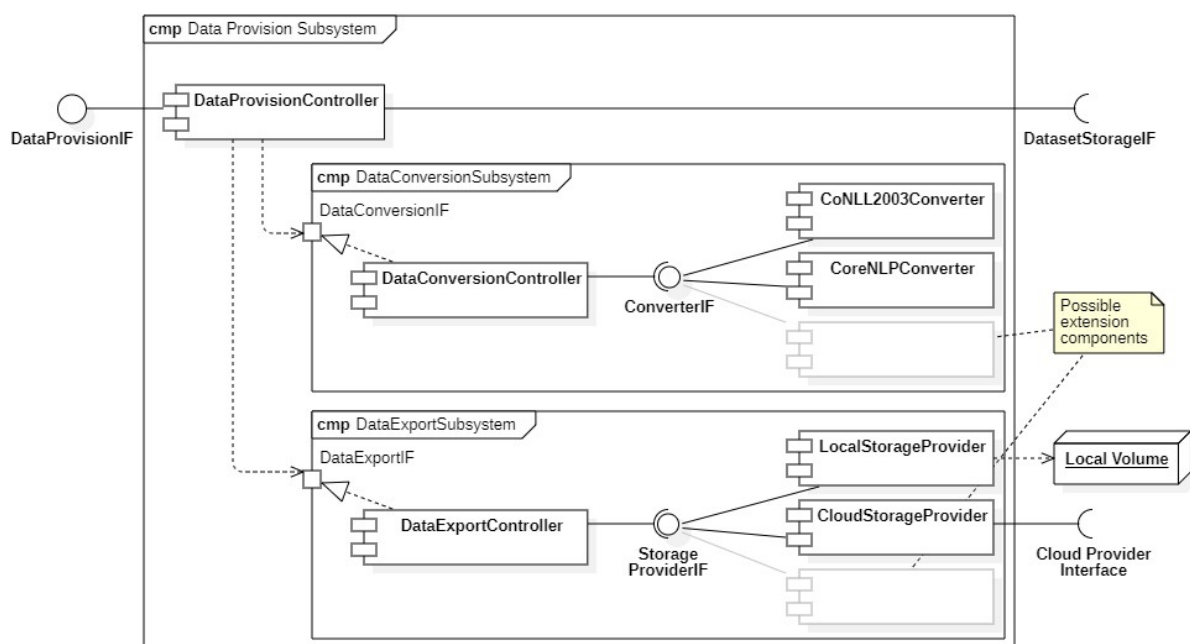
“strategies,” define their behavior through interfaces. In a specialized application of this pattern, the strategies are designed as microservices, each implementing a well-defined HTTP interface. Both strategies and their clients are microservices that allow the selection of a specific strategy based on the context. This adoption of the pattern is illustrated by the presence of empty components in the component diagrams (Figures 2 and 3), which means that additional strategies can be seamlessly integrated into the system without requiring modifications to other components.

The component “Data Preparation Subsystem” encompasses three subsystems: the “Document Import Subsystem,” responsible for parsing and converting documents from various formats into the BioC format; the “Corpus Import Subsystem,” which performs similar tasks for existing corpora; and the “Dataset Storage” subsystem. The “Dataset Storage” subsystem has three primary responsibilities: storing BioC documents for future use, partitioning collections of documents (datasets) into training and test documents, and managing sets of annotation labels for manual document annotation facilitated by a graphical annotation tool. This tool is integrated through the Front-end subsystem within the “Data Preparation” component. A “DataPreparationController” component delegates tasks to the appropriate subsystems and provides an interface for other system components. By employing this extensible architecture utilizing the strategy pattern, the first RC is effectively addressed and resolved.

The second primary component, the “Data Provision Subsystem,” consists of two subsystems: the “Data Conversion Subsystem” and the “Data Export Subsystem.” Both subsystems take advantage of the specialized strategy pattern. The “Data Conversion” subsystem manages the conversion of annotated BioC documents to annotation formats used by various NER frameworks. As new frameworks emerge, there may be a need to support additional formats. In such cases, the system facilitates the seamless integration of new strategies by implementing converters for the new formats. Similarly, the “Data Export” subsystem is responsible for delivering converted documents to various storage solutions, including local file systems and cloud storage services provided by specific cloud service providers. Each storage solution is associated with a dedicated component that serves as a concrete strategy. New storage options can be integrated by implementing new storage providers. For example, the “CloudStorageProvider” serves as a placeholder for specific cloud storage services such as Amazon S3 or Microsoft Azure. Once again, the system’s ability to extend with additional strategies without impacting other parts of the system effectively resolves the second RC.



**Figure 2.** “Data Preparation Subsystem” with subsystems for document- and corpus-import, storage, and user interface. Empty components illustrate the extensibility of the system.



**Figure 3.** Subsystems and components of the “Data Provision Subsystem” with connections to diverse storage solutions. Empty components illustrate the extensibility achieved by incorporating the strategy pattern.



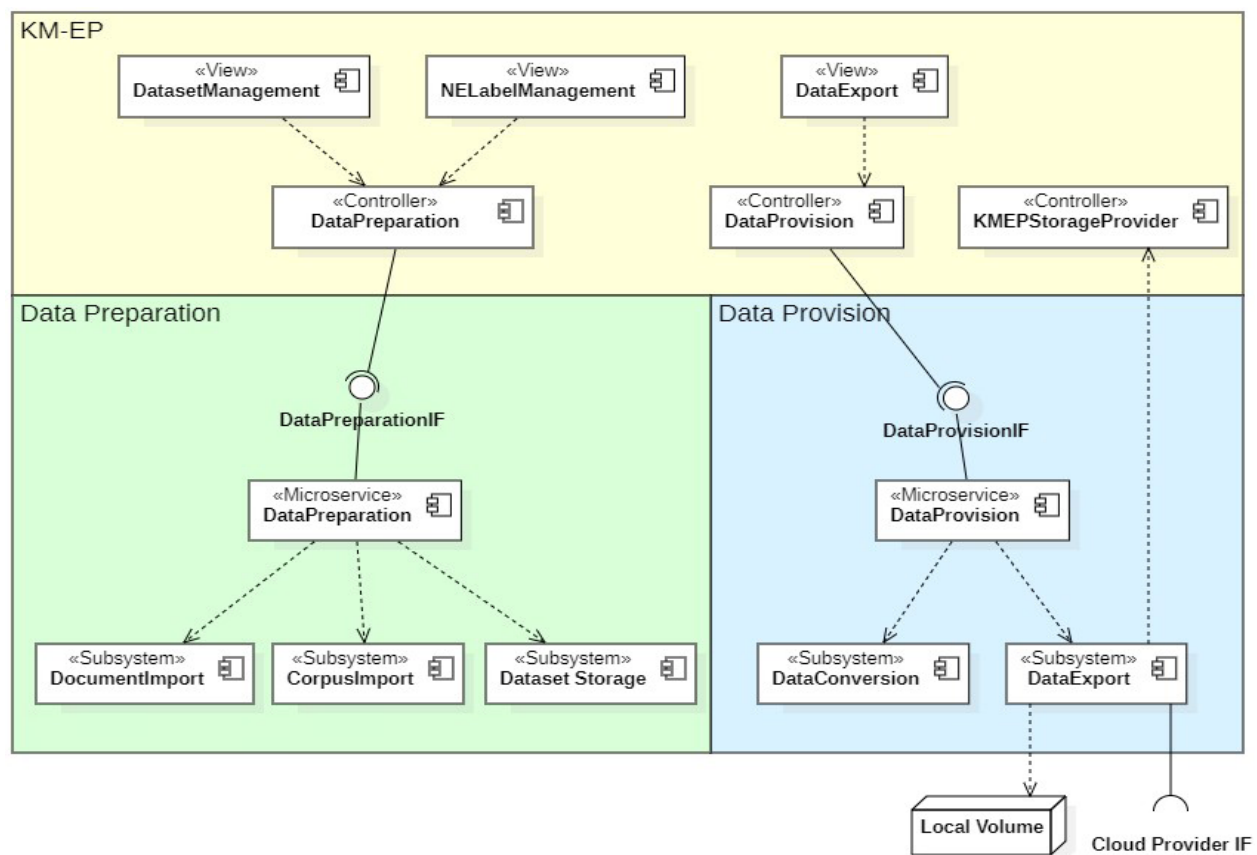


Figure 4. Architecture overview of Prepare4NER integration into KM-EP

The system's graphical user interface (GUI) is integrated within KM-EP, a platform also utilized by the FIT4NER and CIE projects. Figure 4 provides an architectural overview of Prepare4NER, illustrating the two primary components: "Data Preparation" and "Data Provision," including the GUI integration within the KM-EP system. Each component is further subdivided into subsystems to achieve a modular architecture. The subsystems and components specific to "Data Preparation" are depicted in Figure 2, while Figure 3 illustrates those for "Data Provision."

This section has established the theoretical framework for this work. First, the relevant use cases were presented in a use context diagram (Figure 1), and then the use cases were transformed into component diagrams for the "Data Preparation Subsystem" (Figure 2) and the "Data Provision Subsystem" (Figure 3). Finally, the architecture overview (Figure 4) showed how the subsystems are integrated into KM-EP. The following section will now explain the prototypical implementation of the models and their integration into KM-EP in more detail.

## 4. Implementation

This section covers the system development phase and includes RO3: "Development of a prototype for a system that enables the creation of training and test data by medical experts." It details the implementation of the components within the Prepare4NER system modeled in the previous section. Subsequently, these components are integrated into the KM-EP ecosystem to address the system development objectives and facilitate the creation of training and test data by medical experts. All components were implemented as microservices using Java or Python, with integration into KM-EP accomplished using PHP and HTML. The microservice architecture was selected to achieve modularity and scalability, which is essential to implement the strategy pattern discussed earlier. Java and Python were chosen for their widespread adoption in ML and NER, along with the availability of relevant libraries. Various frameworks and libraries were employed to achieve the desired functionality: Spring Boot<sup>55</sup> for implementing HTTP endpoints, managing configurations, and handling database access; FastAPI<sup>56</sup> for creating Python-based

HTTP endpoints; Apache Tika<sup>57</sup> and pdfminer.six<sup>58</sup> for format transformations; and scikit-learn<sup>59</sup> for splitting document sets into training and test data.

For maintainability, all components adhere to a consistent design pattern by implementing a uniform HTTP interface and following a layered architecture. Within the system, controller components such as “DocumentImporterController” and “CorpusImporterController” (refer to Figure 2) in the Data Preparation subsystem, and “DataConversionController” and “DataExportController” (refer to Figure 3) in the “Data Provision Subsystem,” utilize the strategy pattern to access concrete strategies. The implementation of the strategy pattern in the “DataExportController,” depicted in Listing 2, showcases how the specific strategy is chosen based on the request context.

To expand Prepare4NER to support additional document formats, the development of a new Document Converter is essential. This converter is responsible for both the import of new document formats, such as LaTeX and Markdown, and the export of annotated data into annotation formats required by new NER frameworks, such as the spaCy JSON format. Prepare4NER was successfully extended with a converter that provides data in the spaCy JSON format.<sup>60</sup> The development effort for such an extension was found to be limited. The implementation of new converters can be further enhanced by providing a formal description of the interfaces, e.g., by using OpenAPI.<sup>61</sup> Such a description can also outline possible error cases and help implementers of new components handle failures and edge cases, such as corrupted or incomplete data. Proper handling of the described failure states in the core components of the

system would improve the fault tolerance of the overall system. This will be addressed in future enhancements to the prototype described here. The need for formal interface descriptions and clear user-facing error reports was identified during the system evaluation (see Section 5.2).

In terms of official data protection regulations, components such as document converters in the “Document Import Subsystem,” which handle sensitive data such as electronic health records, should also be responsible for anonymization or pseudonymization of the data (see Section 2). Official guidelines, such as those described by the relevant authority<sup>62</sup>, can help implementers meet these data protection requirements. Incorporating data protection into the software development life cycle will help further ensure compliance with the official regulations.<sup>63</sup> These considerations are also subject to future work when the system is deployed in real-world scenarios.

To deploy the system in the cloud, an infrastructure adaptation is necessary to ensure both scalability and security. The Prepare4NER prototype currently uses Docker Compose<sup>64</sup> for microservice orchestration. Therefore, a crucial step in this process is migrating from Docker Compose to Kubernetes by converting the existing docker-compose file into a Helm Chart.<sup>65</sup> Tools such as Kompose<sup>66</sup> can automate this conversion process. To enhance application scalability, Kubernetes features such as autoscaling and load balancing can be utilized. Here, the above-described formal description of interfaces helps to ensure robust inter-service communication. Special attention must be paid to the implementation of security measures, including network policies, secrets management, and identity and access management specific

```
public void exportData(StoreDocumentRequest request) throws
UnknownStorageProviderException { storageProviders.getUrls().
stream()
    .filter(url -> Objects.equals(callGetInfo(url)
        .getStorageProviderId(),
            request.getStorageProviderId()))
    .findFirst()
    .ifPresentOrElse(
        url -> callExportData(url, request),
        () -> new UnknownStorageProviderException(
            "StorageProvider with id " + request.getStorageProviderId() + " not
            supported!")
    );
}
```

**Listing 2.** Implementation of the Strategy Pattern in DataExportController.

to the cloud platform, to ensure data security, compliance, and adherence to data protection regulations, especially in the medical domain. Managed Kubernetes services such as Amazon Elastic Kubernetes Service<sup>67</sup>, Google Kubernetes Engine<sup>68</sup>, or Azure Kubernetes Service<sup>69</sup> offer simplified management and integration. Additionally, infrastructure-as-code tools such as Terraform<sup>70</sup> can be used to automate infrastructure deployment and management. Data consistency, as well as fast access to large datasets to reduce latency, can be achieved by using Kubernetes persistent volumes<sup>71</sup> with storage solutions such as container-attached storage with OpenEBS<sup>72</sup>, or Rook.<sup>73</sup> An experiment evaluating this deployment approach is planned as future work.

To integrate the developed system with KM-EP, Symfony controllers written in PHP, along with Twig

templates, were implemented. These controllers manage tasks by making microservice requests to the appropriate subsystems. The system's data model was replicated using PHP classes. Figure 5 illustrates how the “Data Provision Subsystem” was integrated into the model-view-controller architecture of KM-EP. This approach enabled seamless incorporation of the microservice infrastructure into KM-EP's PHP system, facilitating the use of KM-EP as a storage provider for the “Data Provision Subsystem” (refer to Figure 4). The GUI integration includes an overview of available datasets and controls for importing and creating new datasets, as depicted in Figure 6.

## 5. Evaluation

Section 4 detailed the implementation of a prototype aimed at helping medical experts annotate texts for use

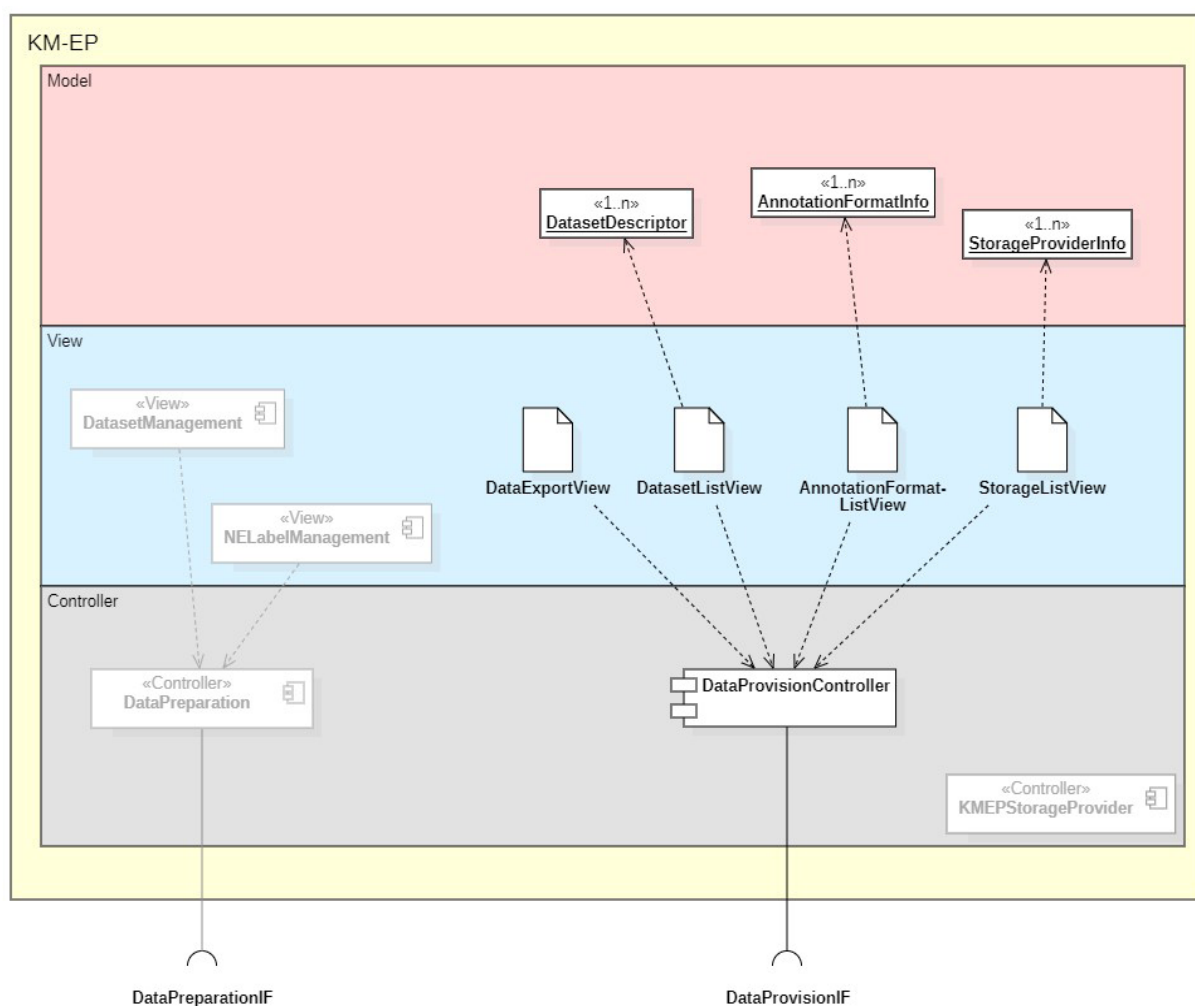
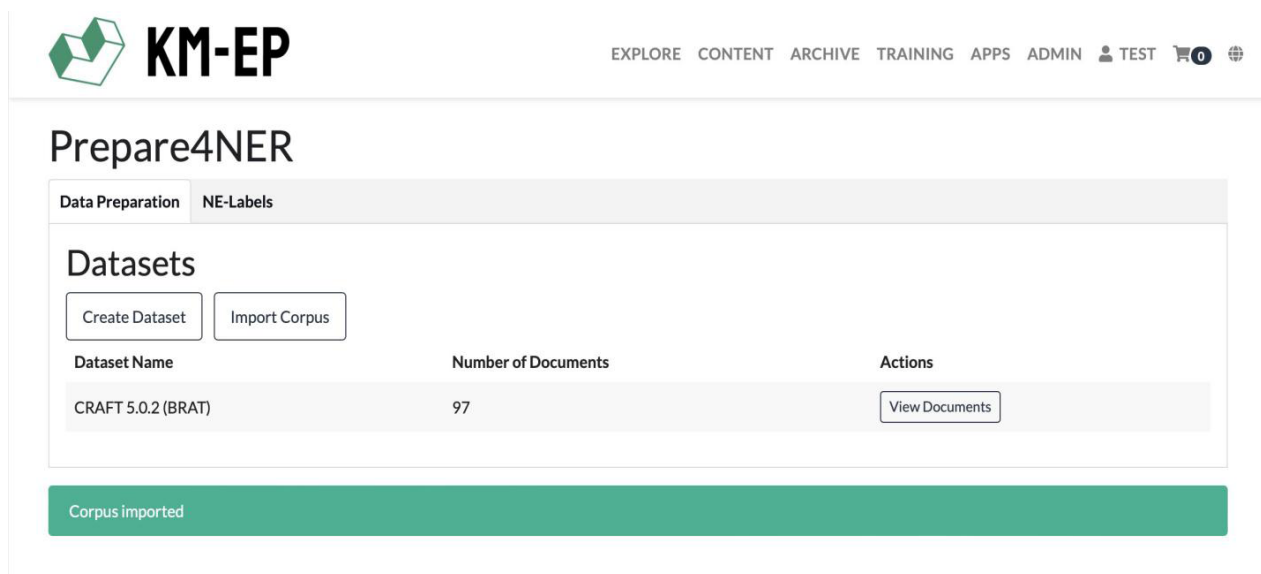


Figure 5. KM-EP Integration of the Prepare4NER “Data Provision Subsystem”



**Figure 6.** Integration of “Dataset Overview” into KM-EP within the “Data Preparation Subsystem”

with different NER frameworks. This section focuses on the experimentation phase, addressing RO4: “Execution and description of expert tests based on the prototype developed.” It first introduces the evaluation methodologies used and then assesses the efficacy of the prototype in supporting medical experts in text annotation across various NER frameworks through expert evaluations and quantitative experiments.

### 5.1. Methodology

This section describes the evaluation methodologies used to assess the system. This work emphasized qualitative methods and synthesized the quantitative experiments reported by Tamla *et al.*<sup>60</sup>. Prepare4NER has been leveraged to streamline data preprocessing for cloud-based NER model training and to conduct additional experiments and quantitative analyses.<sup>60</sup> Future work includes conducting a detailed study and assessing the system’s scalability with large datasets in the cloud. Two distinct qualitative approaches were utilized: Cognitive Walkthrough (CW), a method to evaluate user interfaces outlined by Polson *et al.*<sup>74</sup>, and the Software Technical Review (STR) process<sup>75</sup> is an approach for improving software product quality and is standardized in the Institute of Electrical and Electronics Engineers (IEEE) standard (IEEE Std 1028-2008).<sup>76</sup> CWs involve simulating and analyzing the cognitive processes of a user interacting with the interface. In the initial phase, tasks with specific objectives and required action sequences are defined. Subsequently, a panel of experts assesses the anticipated user interaction with the interface, considering criteria such as action availability and labeling, likelihood

of correct action selection, and action complexity. On the other hand, the STR process is a technique to analyze the technical aspects of a software product. A lead engineer organizes an examination meeting, describing structured inputs containing information such as the software product, review goals, planned functionality, and review process description. During the examination meeting, a group of experts scrutinizes the software product, documenting all findings and anomalies.

### 5.2. Qualitative evaluation findings and recommendations

Two meetings were held for the initial CW evaluation, attended by one researcher with a PhD, two PhD candidates, and one bachelor’s student, all conducting research in the fields of natural language processing (NLP) and NER. In the future, further CW sessions will be planned with medical professionals to evaluate usability outside the field of computer science. For these meetings, seven tasks (see Table 2) were prepared, which covered the entire workflow from importing, annotating, and splitting documents in the Data Preparation phase to exporting datasets for a specific NER framework and storage provider in the “Data Provision” phase.

To explore the flexible architecture of Prepare4NER, the CRAFT corpus was selected for the CW evaluation.<sup>31</sup> It consists of 97 articles from the PubMed Central Open Access subset, each meticulously annotated across various dimensions, including structural layout, coreference, and concept annotation, tailored for ML-based NER.<sup>31</sup> The corpus can be easily converted to the BRAT<sup>26</sup> format

and imported using the “BRATConverter” component (Figure 2). CoNLL-2003<sup>25</sup> was selected as the output format due to its widespread use and support by numerous NER frameworks, such as spaCy, NLTK, Flair, Stanford CoreNLP, and OpenNLP. Although CoNLL-2003 does not represent relations, nested entities, or discontinuous spans, it was deliberately selected here as an interoperability export for common sequence-labeling pipelines; richer structures remain representable in BioC internally and can be exported via alternative converters when required.

Figure 7 illustrates Task T1.3, the import of the CRAFT corpus<sup>31</sup> in BRAT<sup>26</sup> format, version 5.0.2. After a successful import, the list of corpora available in the system is displayed (see Figure 6). To perform Task T1.5, annotation, users click “View Documents” on the selected corpus to see the list of documents contained within the corpus (Figure 8). Clicking “Edit Document” opens the document

editor interface (Figure 9), allowing the user to adjust the annotations. Next, Task T1.6 can be carried out: By clicking the “Split Dataset” button (Figure 8), the function to split the dataset is initiated (Figure 10), the percentage of training documents is entered, and the “Split Dataset” button is clicked. Finally, Figure 11 shows the interface for performing Task T1.7, which prepares the selected corpus in the required format of the chosen NER framework in the selected storage system.

During the evaluation, usability findings were discussed and documented, resulting in a total of 12 findings (Table 3). A significant finding was CWF3, proposing the inclusion of user instructions. Attendees of the CW sessions identified the need for instructions or help text for several tasks to help users perform them more effectively. Similarly, finding CWF5 suggested simplifying the creation of label collections and labels, which was raised in both CW meetings. Furthermore, support for additional annotation features was emphasized (CWF6), as the current state of the annotation tool used does not support discontinuous, nested, or overlapping annotations. Due to current constraints, it was not possible to assess metrics such as the time required to complete various tasks (e.g., time-to-annotate) in this evaluation. However, these metrics are important for understanding how effectively Prepare4NER helps users create training data for ML-based NER models. This analysis is planned for future work. Although these findings identified areas for future improvement, the system was deemed suitable for the intended tasks and addressed all identified RCs.

**Table 2. Tasks for the Cognitive Walkthrough**

#	Task
T1.1	Create a dataset
T1.2	Import a document
T1.3	Import a corpus
T1.4	Create label collections and labels for NERs
T1.5	Annotate a document
T1.6	Split dataset
T1.7	Provide a dataset for a specific NER framework and storage provider

The screenshot displays the 'Prepare4NER' web application interface. At the top, there is a navigation bar with links: EXPLORE, CONTENT, ARCHIVE, TRAINING, APPS, ADMIN, TEST, and a search icon. The main header shows the 'KM-EP' logo and the title 'Prepare4NER'. Below the header, there are two tabs: 'Data Preparation' (selected) and 'NE-Labels'. The 'Import Corpus' section is active, showing a form with the following fields: 'Corpus name' (filled with 'CRAFT 5.0.2 (BRAT)'), 'Corpus source' (filled with 'CRAFT Corpus / Version 5.0.2 / Imported via BRAT Format'), and 'Choose File \*' (filled with 'Craft\_5.0.2\_brat.zip' and a 'Browse' button). At the bottom of the form are two buttons: 'Import Corpus' and 'Cancel'.

**Figure 7.** KM-EP View for Task T1.3: Import a corpus

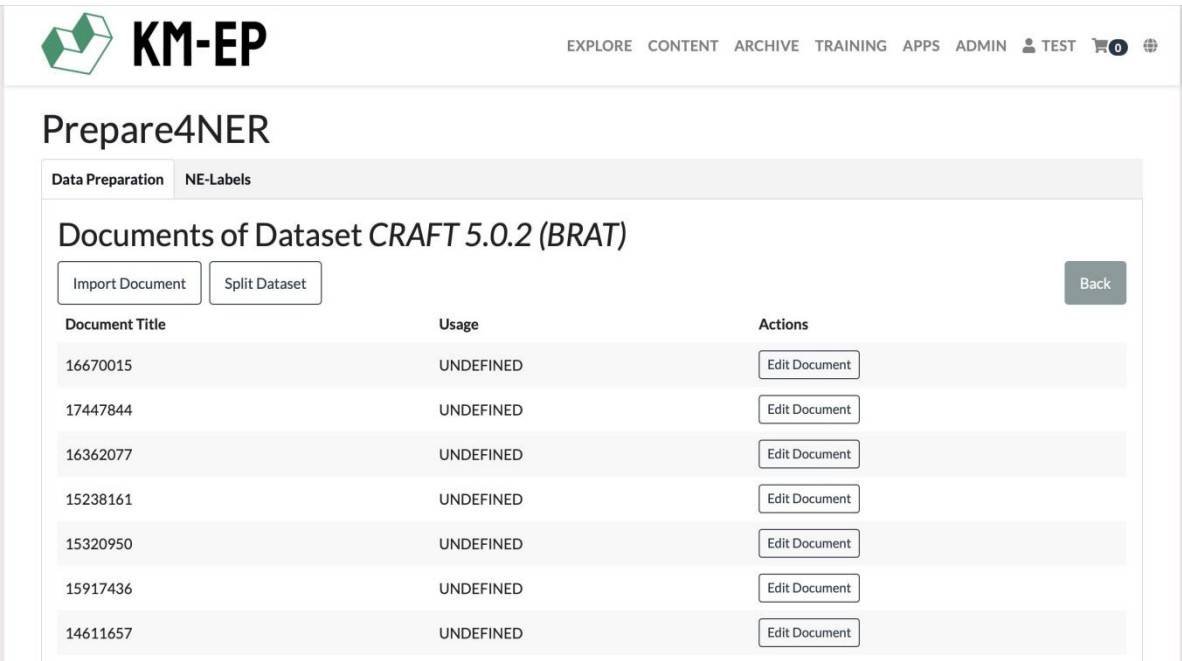


Figure 8. KM-EP View: List of documents

The STR was conducted in a single examination meeting with five input sets (see Section 5.1) listed in Table 4. The meeting included a PhD candidate with expertise in modeling microservice-based architectures and a bachelor’s student in computer science. The outcome of the technical review of the Prepare4NER prototype indicated that the system and its architecture were suitable for the intended purpose. In particular, the strategy pattern facilitates the intended extension of the system for future document formats, NER frameworks, and storage solutions.

In total, 14 findings were documented (Table 5). These findings primarily propose improvements such as better documentation of the HTTP interface of the strategy components (Findings STRF3 and STRF5), allowing system developers to more easily implement new components. Additionally, there is a proposal for a self-registering approach of new components (Finding STRF4) to avoid the need for changing the system configuration when new strategies are attached. Finding STRF8 suggests a system enhancement that allows system developers to test new components before integrating them into the system. In addition, minor findings were documented for future work.

5.3. Quantitative evaluation

This section summarizes the quantitative evaluation experiments conducted with Prepare4NER.<sup>60</sup> The evaluation assesses the effectiveness of Prepare4NER as

a preprocessing and data provisioning component for training ML-based NER models in a cloud environment. All experiments were conducted using the Colorado Richly Annotated Full Text (CRAFT) corpus, which was converted to the BRAT format, imported via Prepare4NER, and exported using the spacyJSONConverter component to generate training data for spaCy-based NER pipelines. Three quantitative experiments were performed, each focusing on a different aspect of the end-to-end workflow. In the first experiment, the impact of available computational resources on model performance was examined by training identical NER models on different Azure compute configurations. Models trained on more powerful instances achieved higher F-scores, while training on less capable machines failed or resulted in reduced performance, demonstrating that sufficient cloud resources are a prerequisite for effective NER model training and that Prepare4NER reliably supports scalable cloud-based execution. The second experiment compared multiple Transformer-based architectures—BERT, GPT-2, ClinicalBERT, and RoBERTa—trained on the Prepare4NER-processed CRAFT corpus. The results, summarized in Table 6, show that all Transformer models outperformed the previously reported baseline results of Furrer *et al.*<sup>77</sup>, where the strongest reference result achieved an F-score of 0.7700 (BERT IDs + spans + OGER). In contrast, the Prepare4NER-based models achieved substantially higher F-scores, including 0.8049 for BERT, 0.8218 for GPT-2, 0.8208 for ClinicalBERT, and



Table 3. Evaluation Results of the Cognitive Walkthrough

Finding	Task	Result
CWF1	T1.1	Integrate the system into existing applications of KM-EP
CWF2	T1.1	Show a list of supported document formats
CWF3	T1.1, T1.2, T1.6	Show instructions
CWF4	T1.2	Enable deletion of documents
CWF5	T1.4	Simplify label collection creation
CWF6	T1.5	Support additional annotation features
CWF7	T1.6	Allow creation of reusable split configurations
CWF8	T1.2	Simplify document import
CWF9	T1.2	Show and explain encountered errors
CWF10	T1.5	Show status of annotation process
CWF11	T1.4	Research the effect of changed labels
CWF12	T1.7	Show progress of export process

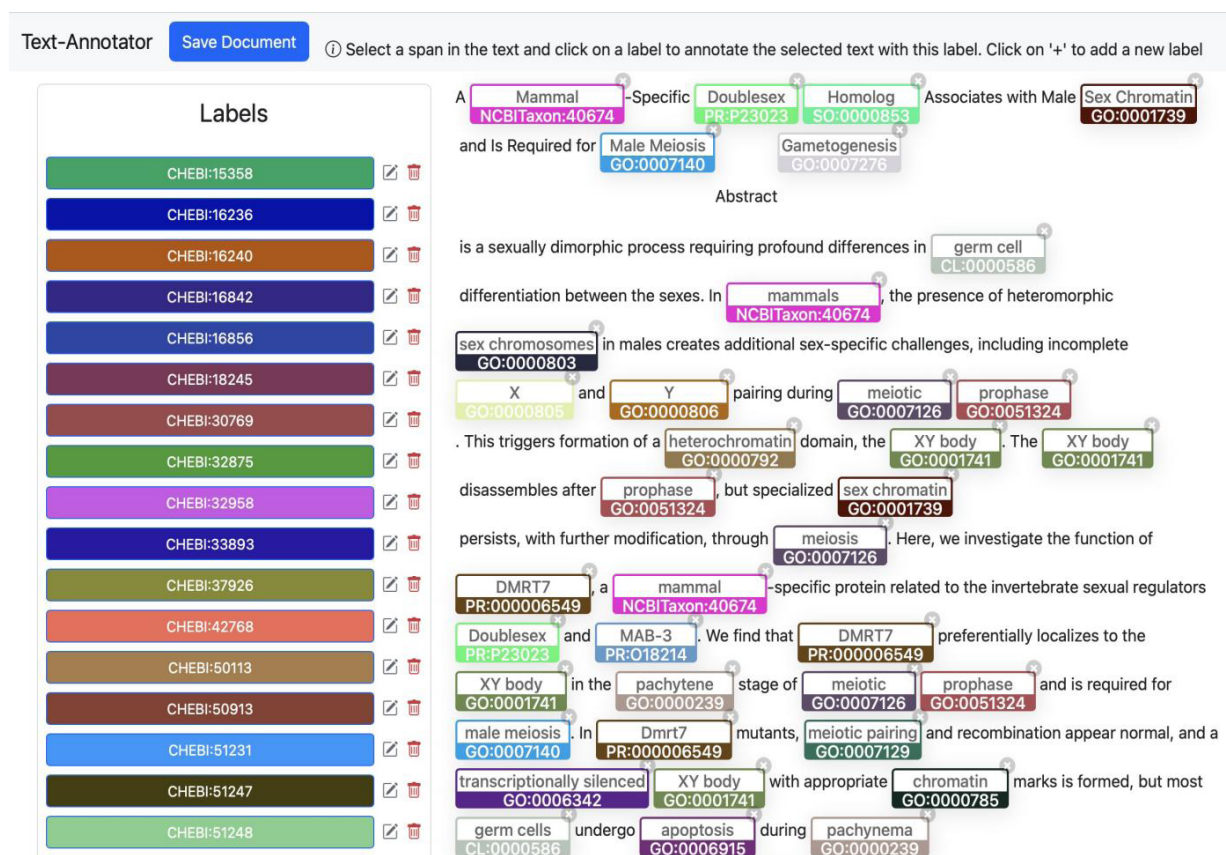
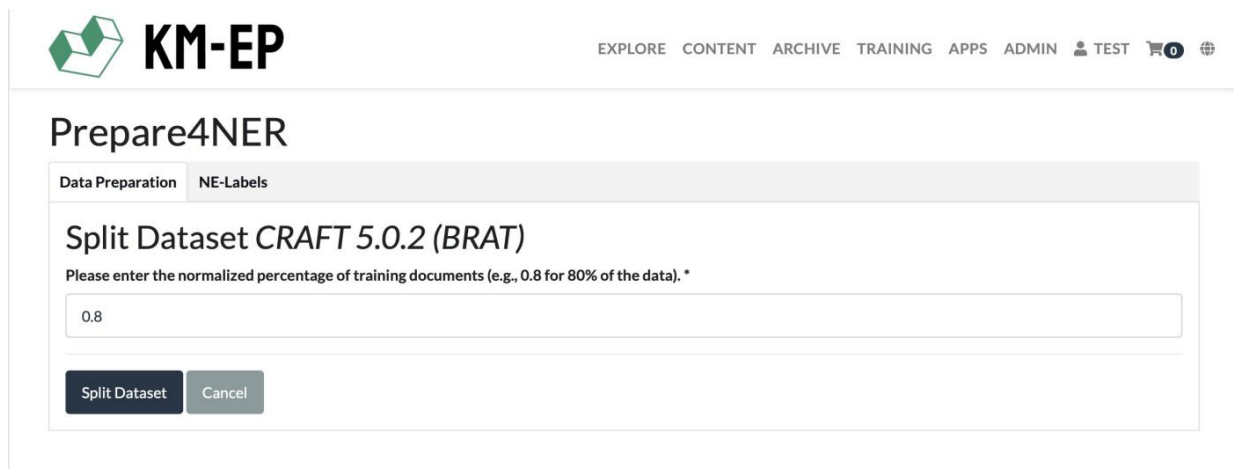
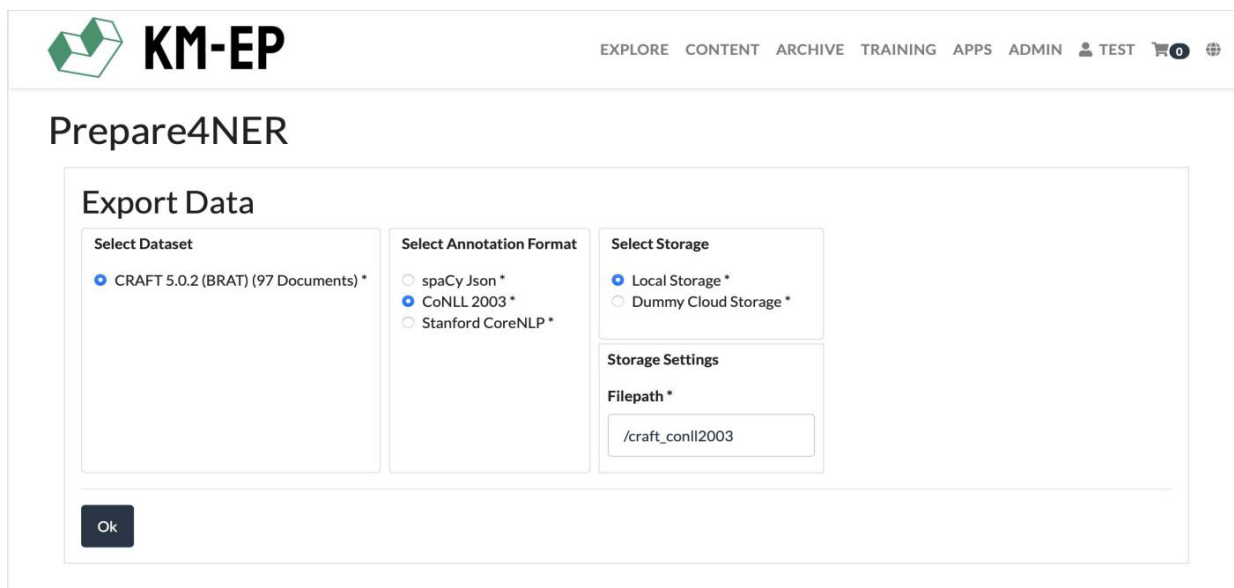


Figure 9. Document-editor UI for Task T1.5: Annotate a document



The screenshot shows the KM-EP web application interface. At the top, there is a navigation bar with links: EXPLORE, CONTENT, ARCHIVE, TRAINING, APPS, ADMIN, TEST, and a shopping cart icon. The main header displays the KM-EP logo. Below the header, the page title is "Prepare4NER". There are two tabs: "Data Preparation" (active) and "NE-Labels". The main content area is titled "Split Dataset CRAFT 5.0.2 (BRAT)". Below this title, a message reads: "Please enter the normalized percentage of training documents (e.g., 0.8 for 80% of the data). \*". A text input field contains the value "0.8". At the bottom of the form, there are two buttons: "Split Dataset" and "Cancel".

Figure 10. KM-EP View for Task T1.6: Split dataset.



The screenshot shows the KM-EP web application interface. At the top, there is a navigation bar with links: EXPLORE, CONTENT, ARCHIVE, TRAINING, APPS, ADMIN, TEST, and a shopping cart icon. The main header displays the KM-EP logo. Below the header, the page title is "Prepare4NER". The main content area is titled "Export Data". There are three main sections: "Select Dataset", "Select Annotation Format", and "Select Storage". In the "Select Dataset" section, "CRAFT 5.0.2 (BRAT) (97 Documents) \*" is selected. In the "Select Annotation Format" section, "CoNLL 2003 \*" is selected. In the "Select Storage" section, "Local Storage \*" is selected. Below these sections, there is a "Storage Settings" section with a "Filepath \*" field containing the value "/craft\_conll2003". At the bottom of the form, there is an "Ok" button.

Figure 11. KM-EP View for Task T1.7: Provide a dataset for a specific NER framework and storage provider

Table 4. Inputs for the Software Technical Review

#	Input
I1	Add support for additional document formats
I2	Add support for additional NER frameworks
I3	Provision training and test data in a cloud environment
I4	Evaluate cloud readiness of the system
I5	Evaluate the integration of the system into KM-EP

a maximum of 0.8337 for RoBERTa, indicating a clear performance gain enabled by the proposed preprocessing and data provisioning workflow.

In the third experiment, the effect of learning-rate optimization on the RoBERTa model was investigated. Different learning-rate configurations were evaluated, demonstrating that appropriate hyperparameter tuning further improves model quality. The optimized RoBERTa configuration achieved the highest overall F-score of 0.8337, exceeding all reported reference results and confirming that Prepare4NER supports not only robust

Table 5. Evaluation results of the Software Technical Review

Finding	Input	Result
STRF1	I1	Implement a more efficient way to identify the correct converter component
STRF2	I1	Enable user to select a converter component
STRF3	I1	Specify the converter interface in the OpenAPI format
STRF4	I2	Let components register themselves
STRF5	I2	Specify the converter interface in the OpenAPI format
STRF6	I3	Documentation of storage provider properties
STRF7	I3	Show error messages in the user interface
STRF8	I3	Provide testing facilities for storage providers
STRF9	I3	Provide definition of meta-data for training and test data
STRF10	I4	Provide a centralized management of the components
STRF11	I4	Monitor components in a cloud environment
STRF12	I4	Reorganize the docker compose file
STRF13	I5	Configure component endpoints inside KM-EP
STRF14	I5	Implement error handling for HTTP calls from KM-EP

Table 6. F-Score results of quantitative experiments

Reference	Experiment	F-Score
Furrer <i>et al.</i> <sup>77</sup>	OGER (baseline)	0.5808
Furrer <i>et al.</i> <sup>77</sup>	BiLSTM no-pretraining	0.7293
Furrer <i>et al.</i> <sup>77</sup>	BiLSTM pretraining	0.7412
Furrer <i>et al.</i> <sup>77</sup>	BiLSTM pick-best	0.7442
Furrer <i>et al.</i> <sup>77</sup>	BERT IDs	0.7555
Furrer <i>et al.</i> <sup>77</sup>	BERT spans+OGER	0.6586
Furrer <i>et al.</i> <sup>77</sup>	BERT IDs+spans+OGER	0.7700
Tamla <i>et al.</i> <sup>60</sup>	BERT	0.8049
Tamla <i>et al.</i> <sup>60</sup>	GPT-2	0.8218
Tamla <i>et al.</i> <sup>60</sup>	ClinicalBERT	0.8208
Tamla <i>et al.</i> <sup>60</sup>	RoBERTa	0.8337

preprocessing but also systematic model optimization. Overall, the combined results of the three experiments demonstrate that Prepare4NER (i) enables efficient cloud-based training, (ii) supports high-performance NER modeling across different Transformer architectures, and (iii) facilitates reproducible performance optimization. At the same time, its modular architecture allows seamless transfer to other knowledge domains and cloud providers, effectively avoiding vendor lock-in.

#### 5.4. Evaluation summary

After conducting expert-based evaluation experiments, it can be inferred that the developed prototype system

demonstrates a fundamental capability to assist medical experts with the annotation of texts for various ML-based NER frameworks. The quantitative experiments show that Prepare4NER both supports the use of powerful cloud resources and enables efficient, cross-domain, high-performance NER modeling through its flexible, modular architecture. Although several findings remain to be addressed in future work and ongoing system development, the achievement of the experimentation objectives confirms the fulfillment of the RQs defined in **Section 1**. The final section will provide a comprehensive summary of all sections of this work and provide a conclusive discussion of the results.

## 6. Conclusion

This article introduced a system designed to streamline the preparation and provision of high-quality annotated data for training ML models in NER. Using the structured Nunamaker methodology for the development of information systems, this system seeks to simplify the creation and maintenance of comprehensive training and test datasets for medical professionals across multiple text formats and for different NER frameworks. Section 1 provided an overview of the topic and positioned the research within its context, setting the stage for the subsequent analysis. In Section 2, we conducted a comprehensive review of the current state of the art, identifying the challenges encountered by medical experts in text annotation and deriving the RCs addressed in this work, thus fulfilling observation objectives. Section 3 presented and discussed design and conceptual models aimed at supporting medical experts in text annotation for ML-based NER, concluding the theory-building objectives and laying the groundwork for Section 4. In Section 4, we detailed the development of a prototype system from the proposed models and demonstrated its integration into the KM-EP system, thereby achieving the system development objectives. Section 5 evaluated the extent to which the developed components addressed the RQs. Expert-based and quantitative evaluation experiments indicated that the prototype system effectively supports medical experts in annotating texts for various ML-based NER frameworks, thereby achieving the experimentation objectives. During these experiments, areas for future work were identified, including improving the documentation of the HTTP interface for strategy pattern components and enabling self-registration of such components (STRF3-5), along with improvements in user documentation (CWF3).

In conclusion, this work successfully addressed the defined RQs and resolved the outstanding challenges identified in Section 2. The findings underscore the potential of the prototype to enhance the NER annotation process for medical professionals and highlight opportunities for further refinement. Prepare4NER is a component of the broader research initiative FIT4NER, which aims to optimize the entire workflow to develop ML models for NER by domain experts. As part of this project, Prepare4NER will be further developed in future research, with a focus on enhancing the robustness of the system and implementing the improvements identified during the evaluations in Section 5. A manuscript reporting the NERFlow study is currently being prepared for publication. The study aims to support the automated annotation of unannotated texts for NER, thereby reducing the annotation burden for medical experts.<sup>39,78</sup> These experts can then primarily

use Prepare4NER to correct automatically annotated data. Future research within FIT4NER will also include a detailed study to evaluate Prepare4NER orchestration in the cloud and its scalability when processing large datasets. In addition, further sessions are planned with medical professionals to evaluate the effectiveness of Prepare4NER in generating training data for ML-based NER models, including relevant metrics. Moreover, the transferability of Prepare4NER to other knowledge domains, such as historical research, will be explored. The contributions of this work lay a solid foundation for advancing the field of NER in medical informatics.

## Acknowledgments

None.

## Funding

None.

## Conflict of interest

The authors declare they have no competing interests.

## Author contributions

*Conceptualization:* Florian Freund, Philippe Tamla

*Formal analysis:* Florian Freund, Philippe Tamla, Matthias Hemmje

*Investigation:* Florian Freund, Philippe Tamla, Sven Stieber

*Methodology:* Florian Freund, Philippe Tamla, Sven Stieber

*Writing—original draft:* Florian Freund, Sven Stieber

*Writing—review & editing:* Philippe Tamla, Matthias Hemmje

## Ethics approval and consent to participate

Not applicable.

## Consent for publication

Not applicable.

## Availability of data

The CRAFT Corpus was used in this work and obtained from <https://github.com/lhunter-lab/CRAFT> under the CC-BY 3.0 license. Code and additional materials are available from the corresponding author upon reasonable request.

## References

1. Freund F, Tamla P, Hemmje M. Towards improving clinical practice guidelines through named entity recognition: Model development and evaluation. In: Proceedings of the 2023 31st Irish Conference on Artificial Intelligence and Cognitive Science (AICS). 2023:1-8.

- doi: 10.1109/AICS60730.2023.10470480
2. Institute of Medicine (US) Committee on Standards for Developing Trustworthy Clinical Practice Guidelines; Graham R, Mancher M, Miller Wolman D, Greenfield S, Steinberg E, eds. *Clinical Practice Guidelines We Can Trust*. National Academies Press; 2011.  
doi: 10.17226/13058
  3. Byyny RL. The data deluge: the information explosion in medicine and science. *Pharos Alpha Omega Alpha-Honor Med Soc Alpha Omega Alpha*. 2012;75(2):2-5
  4. Klerings I, Weinhandl AS, Thaler KJ. Information overload in healthcare: too much of a good thing? *Z Für Evidenz Fortbild Qual Im Gesundheitswesen*. 2015;109(4):285-290.  
doi: 10.1016/j.zefq.2015.06.005
  5. Tamla P, Hartmann B, Nguyen N, Kramer C, Freund F, Hemmje M. CIE: a cloud-based information extraction system for named entity recognition in AWS, AZURE, and medical domain. In: *Communications in Computer and Information Science*. Springer Nature Switzerland; 2023:127-148.  
doi: 10.1007/978-3-031-43471-6\_6
  6. Konkol IM. *Named Entity Recognition*. PhD thesis. University of West Bohemia; 2015.
  7. Bielefeld University. RATIO: Rationalizing Recommendations (RecomRatio). 2017. Available from: <https://spp-ratio.de/projects/recomratio/> [Last accessed on August 6, 2024].
  8. Nawroth C. *Supporting Information Retrieval of Emerging Knowledge and Argumentation*. PhD thesis. FernUniversität in Hagen; 2020
  9. FTK. *Artificial Intelligence for Hospitals, Healthcare & Humanity (AI4H3)*. FTK e.V. Research Institute for Telecommunications and Cooperation; Internal project proposal; 2020. Unpublished.
  10. Liu F, Chen J, Jagannatha A, Yu H. Learning for biomedical information extraction: Methodological review of recent advances. Published online 2016.  
doi: 10.48550/ARXIV.1606.07993
  11. Hemmje M. Chair of Multimedia and Internet Applications. 2023. Available from: <http://www.lgmmia.fernuni-hagen.de/en.html> [Last accessed on].
  12. FTK. FTK e.V. Research Institute for Telecommunications and Cooperation. 2023. Available from: <https://www.ftk.de/en> [Last accessed on February 25, 2023].
  13. Vu B, Wu Y, Afli H, et al. A metagenomic content and knowledge management ecosystem platform. In: *Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE; 2019:1-8
  14. Donovan R, Healy M, Zheng H, et al. SenseCare: Using Automatic Emotional Analysis to Provide Effective Tools for Supporting. In: *Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE; 2018:2682-2687.  
doi: 10.1109/BIBM.2018.8621250
  15. Tamla P, Freund F, Hemmje M. SNERC: Enhancing Knowledge Management with Named Entity Recognition and Document Classification for Apply Gaming. *Artif Intell Appl*. 2025;3(4):392-407.  
doi: 10.47852/bonviewAIA52023841
  16. Tamla P, Boehm T, Nawroth C, Hemmje M. Towards Semantic Web-Based Information Retrieval to solve Information Overload in an Applied Gaming Ecosystem. *Bull IEEE Tech Comm Learn Technol*. 2015;15(2):12
  17. Tamla P, Böhm T, Gaisbachgrabner K, Mertens J, Fuchs M. Survey: Software Search in Serious Games Development. 2019;2348:155-166
  18. Tamla P, Böhm T, Nawroth C, Hemmje M. What do serious games developers search online? A study of GameDev StackExchange. In: *Proceedings of the 5th Collaborative European Research Conference (CERC 2019)*. CEUR workshop proceedings. CEUR-WS.org; 2019; 2348:131-142
  19. Freund F, Tamla P, Reis T, Hemmje M, Kevitt PM. FIT4NER - Towards a Framework-Independent Toolkit for Named Entity Recognition. In: *Proceedings of the CERC 2023*. Hochschule Darmstadt; 2023:10.  
doi: 10.48444/h\_docs-pub-518
  20. Frei J, Kramer F. GERNERMED: An open German medical NER model. *Softw Impacts*. 2022;11:100212.  
doi: 10.1016/j.simpa.2021.100212
  21. Ghiasvand O, Kate RJ. Learning for clinical named entity recognition without manual annotations. *Inform Med Unlocked*. 2018;13:122-127.  
doi: 10.1016/j.imu.2018.10.011
  22. Wen C, Chen T, Jia X, Zhu J. Medical Named Entity Recognition from Un-labelled Medical Records based on Pre-trained Language Models and Domain Dictionary. *Data Intell*. 2021;3(3):402-417.  
doi: 10.1162/dint\_a\_00105
  23. Giachelle F, Irrera O, Silvello G. MedTAG: a portable and customizable annotation tool for biomedical documents. *BMC Med Inform Decis Mak*. 2021;21(1):352.  
doi: 10.1186/s12911-021-01706-4
  24. Pinto A, Oliveira HG, Alves AO. Comparing the performance of different NLP toolkits in formal and social media text. In: Mernik M, Leal JP, Oliveira HG, eds. *5th Symposium on Languages, Applications and Technologies (SLATE'16)*. OpenAccess series in informatics (OASIs). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik; 2016;51:16.

- doi: 10.4230/OASICS.SLATE.2016.3
25. Sang EFTK, De Meulder F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *arXiv*. Published online 2003.  
doi: 10.48550/arXiv.cs/0306050
  26. Stenetorp P, Pyysalo S, Topić G, Ohta T, Ananiadou S, Tsujii J. BRAT: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. 2012:102-107.
  27. Comeau DC, Islamaj Doğan R, Ciccarese P, *et al*. BioC: a minimalist approach to interoperability for biomedical text processing. *Database*. 2013;2013(0):bat064.  
doi: 10.1093/database/bat064
  28. Stieber S. *Implementierung eines Systems für Vorverarbeitung von Daten für Named Entity Recognition in einem Wissensmanagement-System für den medizinischen Bereich*. Bachelor's thesis. FernUniversität in Hagen; 2023
  29. Nunamaker Jr JF, Chen M, Purdin TDM. Systems Development in Information Systems Research. *J Manag Inf Syst*. 1990;7(3):89-106.  
doi: 10.1080/07421222.1990.11517898
  30. Li J, Sun A, Han J, Li C. A Survey on Deep Learning for Named Entity Recognition. *IEEE Trans Knowl Data Eng*. 2022;34(1):50-70.  
doi: 10.1109/TKDE.2020.2981314
  31. Cohen KB, Verspoor K, Fort K, *et al*. The Colorado Richly Annotated Full Text (CRAFT) Corpus: Multi-Model Annotation in the Biomedical Domain. In: Ide N, Pustejovsky J, eds. *Handbook of Linguistic Annotation*. Springer Netherlands; 2017:1379-1394.  
doi: 10.1007/978-94-024-0881-2\_53
  32. Bada M, Vasilevsky N, Baumgartner WA, Haendel M, Hunter LE. Gold-standard ontology-based anatomical annotation in the CRAFT Corpus. *Database*. 2017;2017:bax087.  
doi: 10.1093/database/bax087
  33. Kittner M, Lamping M, Rieke DT, *et al*. Annotation and initial evaluation of a large annotated German oncological corpus. *JAMIA Open*. 2021;4(2):ooab025.  
doi: 10.1093/jamiaopen/ooab025
  34. Ahmadi S, Shah A, Fox E. Retrieval-based Text Selection for Addressing Class-Imbalanced Data in Classification. *arXiv*. Preprint posted online 2023:arXiv:2307.14899.  
doi: 10.48550/arXiv.2307.14899
  35. Newman-Griffis D, Divita G, Desmet B, Zirikly A, Rosé CP, Fosler-Lussier E. Ambiguity in medical concept normalization: An analysis of types and coverage in electronic health record datasets. *J Am Med Inform Assoc*. 2021;28(3):516-532.  
doi: 10.1093/jamia/ocaa269
  36. Alhassan A, Schlegel V, Aloud M, Batista-Navarro R, Nenadic G. Discontinuous named entities in clinical text: A systematic literature review. *J Biomed Inform*. 2025;162:104783.  
doi: 10.1016/j.jbi.2025.104783
  37. Liang S, Profitlich HJ, Klass M, *et al*. Building A German Clinical Named Entity Recognition System without In-domain Training Data. In: Proceedings of the 6th Clinical Natural Language Processing Workshop. Association for Computational Linguistics; 2024:70-81.  
doi: 10.18653/v1/2024.clinicalnlp-1.7
  38. Nastou K, Koutrouli M, Pyysalo S, Jensen LJ. CoNECo: a Corpus for Named Entity recognition and normalization of protein Complexes. *Bioinforma Adv*. 2024;4(1):vbae116.  
doi: 10.1093/bioadv/vbae116
  39. Freund F, Tamla P, Tran B, Hemmje M. Evaluating NERFlow: User-Centered Assessment of Automated LLM-Based Annotation for Medical Named-Entity Recognition. *Procedia Comput Sci*. In press.
  40. Liu J, Wong ZSY. Utilizing active learning strategies in machine-assisted annotation for clinical named entity recognition: a comprehensive analysis considering annotation costs and target effectiveness. *J Am Med Inform Assoc*. 2024;31(11):2632-2640.  
doi: 10.1093/jamia/ocae197
  41. Artstein R, Poesio M. Inter-Coder Agreement for Computational Linguistics. *Comput Linguist*. 2008;34(4):555-596.  
doi: 10.1162/coli.07-034-R2
  42. Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF. Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research. *Yearb Med Inform*. 2008;17(01):128-144.  
doi: 10.1055/s-0038-1638592
  43. Raja U, Mitchell T, Day T, Hardin JM. Text mining in healthcare. Applications and opportunities. *J Healthc Inf Manag JHIM*. 2008;22(3):52-56
  44. Neves M, Ševa J. An extensive review of tools for manual annotation of documents. *Brief Bioinform*. 2021;22(1):146-163.  
doi: 10.1093/bib/bbz130
  45. Kim JD, Ohta T, Tateisi Y, Mima H, Tsujii J. XML-based linguistic annotation of corpus. In: Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02), 2001:47-53.
  46. Ogren P. Knowtator: A Protégé plug-in for annotated corpus



- construction. In: Proceedings of the Human Language Technology Conference of the NAACL, Companion V. 2006:273-275.
47. Krishnamoorthy S, Jiang Y, Buchanan W, Singh A, Ortega J. CLPT: a universal annotation scheme and toolkit for clinical language processing. In: Naumann T, Bethard S, Roberts K, Rumshisky A, eds. In: Proceedings of the 4th Clinical Natural Language Processing Workshop. Association for Computational Linguistics; 2022:1-9.  
doi: 10.18653/v1/2022.clinicalnlp-1.1
  48. Sharir O, Peleg B, Shoham Y. The cost of training NLP models: a concise overview. arXiv Published online 2020.  
doi: 10.48550/ARXIV.2004.08900
  49. Putzier M, Khakzad T, Dreischarf M, Thun S, Trautwein F, Taheri N. Implementation of cloud computing in the German healthcare system. *Npj Digit Med*. 2024;7(1):12.  
doi: 10.1038/s41746-024-01000-3
  50. Wang H, Wang B, Wang S. Design and Implementation of a Primary Healthcare Cloud Platform. *Front Comput Intell Syst*. 2024;7(3):77-84.  
doi: 10.54097/01kn4y43
  51. Akerele JI, Uzoka A, Ojukwu PU, Olamijuwon OJ. Improving healthcare application scalability through microservices architecture in the cloud. *Int J Sci Res Updat*. 2024;8(2):100-109.  
doi: 10.53430/ijrsru.2024.8.2.0064
  52. Norman DA, Draper SW. *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc.; 1986
  53. Rumbaugh J, Jacobson I, Booch G. *The Unified Modeling Language Reference Manual*. 2nd ed. Addison-Wesley; 2005.
  54. Gamma E, Johnson R, Helm R, Vlissides J. *Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software*. Pearson Deutschland GmbH; 2011
  55. Mane D, Chitnis K, Ojha N. The spring framework: An open source java platform for developing robust java applications. *Int J Innov Technol Explor Eng IJITEE*. 2013;3(2):137-143
  56. Ramírez S. FastAPI. Published online 2023. Available from: <https://fastapi.tiangolo.com/> [Last accessed on October 3, 2023].
  57. The Apache Software Foundation. Apache Tika: a content analysis toolkit. Published online 2023. Available from: <https://tika.apache.org/> [Last accessed on July 24, 2023].
  58. pdfminer community. pdfminer.six: We fathom PDF. Published online 2022. Available from: <https://github.com/pdfminer/pdfminer.six> [Last accessed on July 22, 2023].
  59. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in python. *J Mach Learn Res*. 2011;12:2825-2830
  60. Tamla P, Freund F, Hemmje M. Cloud-based medical named entity recognition: a FIT4NER-based approach. *Information*. 2025;16(5):395.  
doi: 10.3390/info16050395
  61. OpenAPI Specification. Available from: <https://github.com/OAI/OpenAPI-Specification/tree/main> [Last accessed on February 1, 2025].
  62. Federal Ministry for Economic Affairs and Climate Action. Guidelines on the protection of health data. Available from: <https://www.bmwk.de/Redaktion/EN/Dossier/guidelines-on-the-protection-of-health-data.html> [Last accessed on February 1, 2025].
  63. Irvine C, Balasubramaniam Dharini, Henderson T. Short paper: Integrating the data protection impact assessment into the software development lifecycle. In: *Lecture Notes in Computer Science*. Springer International Publishing; 2020:219-228.  
doi: 10.1007/978-3-030-66172-4\_13
  64. Docker Compose[Docker Docs. Available from: <https://docs.docker.com/compose/> [Last accessed on January 29, 2025].
  65. Helm Authors. Helm - The package manager for Kubernetes. 2025. Available from: <https://helm.sh/> [Last accessed on August 25, 2025].
  66. Kompose - Convert your Docker Compose file to Kubernetes or OpenShift. Available from: <https://kompose.io/> [Last accessed on January 29, 2025].
  67. Amazon Elastic Kubernetes Service Documentation. Available from: <https://docs.aws.amazon.com/eks/> [Last accessed on January 29, 2025].
  68. Google Kubernetes Engine (GKE)|Google Cloud. Available from: <https://cloud.google.com/kubernetes-engine> [Last accessed on January 29, 2025].
  69. Azure Kubernetes Service (AKS) documentation | Microsoft Learn. Available from: <https://learn.microsoft.com/en-us/azure/aks/> [Last accessed on January 29, 2025].
  70. Terraform by HashiCorp. Available from: <https://www.terraform.io/> [Last accessed on January 29, 2025].
  71. Nocentino AE, Weissman B. Storing persistent data in kubernetes. In: *SQL Server on Kubernetes: Designing and Building a Modern Data Platform*. Apress; 2021:111-137.  
doi: 10.1007/978-1-4842-7192-6\_6
  72. Container attached storage (CAS). Available from: <https://openefs.io/docs/2.12.x/concepts/cas> [Last accessed on February 1, 2025].
  73. Rook - cloud-native storage orchestrator for Kubernetes. Available from: <https://github.com/rook/rook> [Last accessed on February 1, 2025].
  74. Polson PG, Lewis C, Rieman J, Wharton C. Cognitive

- walkthroughs: a method for theory-based evaluation of user interfaces. *Int J Man-Mach Stud.* 1992;36(5):741-773.  
doi: 10.1016/0020-7373(92)90039-N
75. Collofello JS. The Software Technical Review Process. Published online 1988. Accessed. Available from: <https://web.archive.org/web/20150724025200/http://www.sei.cmu.edu/reports/88cm003.pdf> [Last accessed on May 12, 2020].
76. IEEE. *IEEE standard for software reviews and audits.* IEEE Std 1028-2008. IEEE; 2008.  
doi: 10.1109/IEEESTD.2008.4601584
77. Furrer L, Cornelius J, Rinaldi F. UZH@CRAFT-ST: a Sequence-labeling Approach to Concept Recognition. In: *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks.* Association for Computational Linguistics; 2019:185-195.  
doi: 10.18653/v1/D19-5726
78. Freund F, Tamla P, Tran B, Hemmje M. Open-Source Large Language Models for FIT4NER: Automatic Annotation for Medical Named Entity Recognition. Manuscript submitted for publication. 2025.