Healthcare information system engineering: AI technologies and open source approaches

Shen, Z.

Chapter 8

Conclusions

The purpose of this research is to enhance the development and deployment of healthcare information systems in practice. The interdisciplinarity of HIS engineering research provides us research opportunities from both IT and healthcare perspectives. This dissertation focused on: 1) how to solve specific real-world healthcare problems with HIS; 2) how to employ state-of-the-art of information technology to accelerate HIS engineering. The empirical work of this dissertation demonstrates the usefulness and effectiveness of HIS in practice.

In this concluding chapter, we answer the Main Research Question (MRQ) in section 8.1 by discussing answers to all Research Questions (RQs) as presented in section 1.4. We end by providing a reflection on this dissertation in section 8.2, which includes a discussion of its limitations (subsection 8.2.1), an outlook on future work (subsection 8.2.2), and a personal reflection (**??**).

8.1 Answers to Research Questions

MRQ

The aim of this research is formalised in the following research question:

> *How can we employ artificial intelligence technologies – such as machine learning algorithms, knowledge systems and natural language processing techniques – based on open source principles to accelerate healthcare information system engineering in solving real-world clinical problems?*

Turning to our main research question, we have investigated the application of various AI technologies in healthcare. The dissertation started with the clinical descision support system we developed in chapter 2, where we focused on designing a federated information architecture to integrate various healthcare data sources from different countries. The system supported a large multinational randomized clinical trial over 2 years without any technical issue. However, we also identified a number of improvements to the system, such as enhancing the data entry process with NLP-enabled data extraction. To address this we experimented with an NLP prototype in chapter 3. The prototype built on a proposed lightweight NLP framework which extracted medical entities such as diseases and medications and then transformed these into standardized medical codes. The evaluation of the prototype shows promising results, so there is a potential of using the system to replace data entry in healthcare information systems. In chapter 4, we further explored the use of NLP data extraction for adverse drug reactions from drug product labels. An NLP pipeline was developed to collect drug product labels from Electronic Medicines Compendium (EMC) and then extract adverse drug reactions from them. The evaluation conducted on 647 common medicines shows that the pipeline is capable of extracting adverse drug reactions with high precision and recall. It has the potential of being used to solve ADR related clinical problems, such as creating adverse drug reaction databases (Kuhn et al., 2016; Garcia et al., 2023).

We demonstrated the potential of NLP techniques in extracting information from clinical texts. But all the studies were conducted on small datasets. Given the accumulation of huge amounts of clinical data over last decades, it is worth exploring the use of NLP techniques in extracting information from large-scale clinical data. In chapter 5, we proposed a big data framework for scalable and efficient information extraction from large-scale biomedical and clinical data. This framework enabled us to perform topic modeling on **over 1 million full-text articles** from the PMC Open Access Subset in about 25 minutes at a cost of $1.5 USD on Google Cloud.

With the development of AI, there are so many AI technologies including NLP techniques, algorithms, packages. In chapter 6, a systematic literature review is conducted to identify the state-of-the-art of open source clinical software. In chapter 7, we have developed a web-based application to support clinical developers' decisions in the software development process with a reproducible and scalable method for systematic studies on open source clinical software.

This dissertation has demonstrated how to help accelerate healthcare information system engineering with AI technologies in real-world clinical settings. Both scientific and technical contributions are of great importance in our studies. Besides designing federated information architecture, big data framework and systematic literature review, we also developed NLP pipelines and web-based applications with open source principles. We hope that clinical developers and researchers will reuse our public accessible source codes and data for their research and clinical applications (Tanaka et al., 2024; Garcia et al., 2023).

RQ1 (chapter 2)

> *How can we design a GDPR-compliant clinical decision support system that supports a multi-center multi-lingual clinical trial?*

To answer this question, this chapter designed a federated information architecture that is capable of integrating data from different countries in a unified schema. A clinical decision support system named STRIPA was developed based on the above architecture to support a large multinational randomized clinical trial which requires multilingual support, data security and privacy, data accessibility and consistency. Evaluation of the system was conducted in both development and deployment. Throughout development random manual comparisons of data between databases and data sources were performed to test on data completeness, data consistency and data integrity. During the clinical trial information of 2008 older adults from 110 clusters of inpatient wards within university based hospitals in four European countries were stored and processed. STRIPA supported the clinical trial for 2 years without any technical issue (Blum et al., 2021a). Sensitive patient data in the system were kept secure and private.

From a broader perspective, the success of this clinical decision support system leads us to believe that the federated information architecture and data integration methods have a good potential to be reused in similar scenarios, especially, in large-scale clinical trials.

RQ2 (chapter 3)

> *How can we improve rapid and cost-effective development of clinical NLP systems with external NLP APIs?*

There are various methods and frameworks for developing clinical NLP systems. However, most of them require a tremendous amount of resources to ensure successful implementation (Chiang et al., 2010). In this chapter we presented a lightweight framework which enables a rapid development of clinical NLP systems with external NLP APIs. In addition, a web-based open source clinical NLP application was built to validate the framework and they together answer RQ2.

The lightweight framework consists of four main components: external APIs, infrastructure, NLP pipelines, and Apps. External API refers to cloud-based NLP services which are available for everyone via APIs. The pay-as-you-go billing model offers end-users flexibility and freedom. The infrastructure layer ensures data privacy and security by preparing clinical data with deidentification and authentications before sending them to external APIs. NLP pipelines parse unstructured medical text and map them into structured and standardized medical codes with the external APIs component. By assembling suitable cloud-based NLP services on demand in NLP pipelines, we are able to process unstructured medical text with the least effort. In the application layer, clinical NLP-enabled applications for various needs can be created.

A clinical concept extraction app is created based on the proposed framework in the chaper. It extracts clinical concepts from clinical free texts. Evaluation of the application on four test datasets shows satisfactory overall results, which proves the effectiveness and capabilities of the proposed framework. Moreover less time and resources are required to create and maintain NLP-enabled clinical tools given that all NLP tasks are outsourced.

RQ3 (chapter 4)

> *How can we utilize NLP techniques to automatically extract adverse drug reactions from the summary of product characteristics in European drug product labels?*

In this chapter, we designed an NLP pipeline that automatically scrapes the summary of product characteristics online and then extracts adverse drug reactions from them. It starts with scraping side effects data of 647 commonly used medicines from the Electronic Medicines Compendium (EMC) (Electronic Medicines Compendium (EMC), 2020). The second step of the pipeline is to extract adverse drug reactions from different formats of side effects data. For side effects that are recorded in free-text, named entity recognition (NER) is used. Side effects written in structured text can be easily extracted by syntax parsing.

The manual experts review results demonstrate that our proposed method is effective and has the potential of being used to solve ADR related clinical problems.

Specifically, our method achieves an overall recall of 0.990 and a precision of 0.932.

RQ4 (chapter 5)

> *How can we design an open source big data framework to facilitate cost-effective, large-scale, biomedical literature mining?*

To address this research question, we proposed a big data framework specifically designed for large-scale biomedical literature mining. Our framework consists of three key components: storage layer, management layer and spark apps layer. These components work together on top of a cloud infrastructure layer (Infrastructure-as-a-Service) to provide a scalable, flexible, efficient, and affordable solution for large-scale biomedical literature mining.

To validate the effectiveness of our framework, we conducted two case studies to evaluate the performance of our framework. To begin with, we performed topic modeling on over 1 million full-text articles from the PMC Open Access Subset, demonstrating the framework's scalability by analyzing a much larger dataset than previous studies. It processed over 1 million articles in about 25 minutes at a cost of $1.5 USD, proving significantly faster and more cost-effective than other systems. The second study conducted cancer article classification using 29,437 labeled full-text articles, comparing performance with SparkText. Our framework was 6 times faster, with lower costs and improved prediction results. These case studies effectively showcased the framework's scalability, efficiency, and cost-effectiveness for large-scale biomedical literature mining tasks, outperforming existing solutions in execution time, cost, and prediction performance.

RQ5 (chapter 6)

> *How can we support clinical developers' decisions in the software development process with a reproducible and scalable method for systematic studies on open source clinical software?*

To conduct the systematic studies on open source clinical software, we developed a reproducible and scalable data pipeline to collect and analyze GitHub repository data, providing insights into the current status, popularity factors, and main focus areas of open source clinical software. Our approach offers several key contributions:

- A quantitative overview of the open source clinical software landscape, including growth trends, popular repositories, and geographic distribution.

- Identification of factors impacting repository popularity, such as forks and number of contributors.
- Discovery of 10 main focus areas through topic modeling, revealing the evolving trends in clinical software development.
- A reproducible methodology that can be easily adapted to study open source software in other domains.

By providing these insights and a reusable approach, this study empowers clinical developers to make more informed decisions in their software development process. They can better understand the available open source resources, identify popular and actively maintained projects, and recognize emerging trends in clinical software development. The reproducible methodology in this study also encourages collaboration and continuous improvement in the field, as other researchers can build upon and extend this work to provide even more comprehensive insights in the future.

While limitations exist, such as the focus on GitHub and manual topic labeling, this research lays the groundwork for future studies on open source clinical software. The reproducible nature of our method encourages further refinement and application to support ongoing improvements in clinical software development and reuse.

RQ6 (chapter 7)

> *How can we design an online platform where clinical developers can easily locate and confidently select appropriate open-source clinical software based on associated scientific literature?*

This chapter presents LOCATE, a web application designed to address the challenge of efficiently linking open-source clinical software with relevant scientific literature. The system answers the main research question by:

1. Developing a data pipeline that collects and processes information about open-source clinical software from GitHub and associated literature from Google Scholar.
2. Creating an interactive web application that allows users to search for clinical software and view related papers, providing additional insights beyond typical software documentation.

3. Implementing a peer review cycle that enables continuous improvement of the platform through expert feedback.

4. Offering an open-source, modular architecture that facilitates further development and expansion.

The evaluation demonstrates that LOCATE provides valuable additional knowledge about clinical software through linked papers, enabling more informed decision-making for both researchers and practitioners. While limitations exist, such as relying solely on GitHub for software data, LOCATE represents a significant step towards bridging the gap between open-source clinical software and scientific literature. This work lays the foundation for future research and improvements in this area, ultimately contributing to more effective utilization of open-source tools in clinical research and practice.

8.2 Reflection

This section of the dissertation discusses the Limitations of this work, proposes some future research opportunities, and ends with a personal reflection.

8.2.1 Limitations

We first reflect on the AI technologies used in this dissertation. We then discuss the datasets we used to evaluate the performance of the proposed methods. We continue with discussing the privacy and security issues in HIS. We conclude with elaborating on the impact of LLMs.

AI technologies

AI technologies cover a variety of topics including machine learning, deep learning, NLP and LLM. AI technologies used in this dissertation include traditional NLP techniques like topic modeling and text classification. In chapter 4, chapter 5 and chapter 6, we employed AI technologies to process big clinical data. While these methods demonstrated good performance for their respective tasks, they have some inherent limitations. The NLP techniques used in this dissertation are relatively basic compared to current state-of-the-art approaches, such as deep learning or transformer-based models. For example, chapter 4 uses rule-based parsing techniques specifically designed for SmPC (Summary of Product Characteristics) documents, rather than more generalizable approaches. chapter 5 and chapter 6 rely on traditional topic modeling and text classification methods instead of exploring recent advances in neural architectures that could potentially yield better results. Given the wide adoption of deep learning methods in NLP tasks, some comparison studies would be useful.

Data bias

One key threat to the validity of this dissertation is the data bias. First, most of data used to evaluate the proposed methods and prototypes in this study are in English. For instance, the ADR extraction pipeline in chapter 4 was tailored for English drug products from SmPC. The decision to overlook this issue was not intentional but rather a consequence of broader limitations in clinical data resource availability. Specifically, public medical and clinical datasets in general remain scare, and datasets languages other than English are exceptionally scarce. However, these methods can be adapted for application in non-English clinical data. The big data framework from chapter 5 is applicable to large clinical datasets in any language. Secondly, the open source research in chapter 6 and chapter 7 are purely based on GitHub. Although GitHub serves as the main platforms for open source projects, there are also other platform such as GitLab and Bitbucket.

Privacy and security

Data privacy and security is considered as one of the most important aspects in research using clinical data. Although systems from chapter 3 and chapter 4 included the privacy and security component, this study lacks explicit research focusing on privacy-preserving methodologies and security measures for clinical data storage and transmission. Given the sensitive nature of health information, ensuring patient confidentiality and safeguarding data against breaches are critical in real-world applications. Therefore, privacy and security related rules and guidelines should be carefully examined, and it lays the foundation for the utilization of AI technologies in clinical settings.

The advocacy of open source clinical software in this study has its security limitations. While open source software offers benefits such as cost-effectiveness, transparency, and community-driven innovation, its application in healthcare has some security concerns. First of all, open source software may contain undiscovered vulnerabilities or dependencies on inadequately maintained libraries, which could expose sensitive patient data to breaches if not rigorously audited or updated. Unlike commercial software which has built-in safeguards for compliance with healthcare-specific regulations (e.g., HIPAA, GDPR), open source software needs extra work to meet the compliance requirements. Lastly, security patches and updates depend on community contributions, which may lag behind emerging threats, leaving systems unprotected during critical windows.

Impact of LLMs

This dissertation was completed before the release of ChatGPT. The rapid adoption and advancement of LLMs after ChatGPT have fundamentally transformed the application of clinical natural language processing, making many of the techniques presented in this work obsolete. The powerful capabilities of LLMs in general tasks surpass the rule-based parsing and traditional NLP methods employed in this research. While the approaches introduced in this research were state-of-the-art at the time of their creation, LLMs have introduced a paradigm shift in processing clinical text.

The API-based framework proposed in chapter 3 aligns remarkably well with today's LLM ecosystem, as both rely on external service integration through APIs. This architectural similarity makes the framework highly adaptable to LLM integration. Furthermore, LLMs significantly simplify the framework by consolidating multiple NLP tasks (entity extraction, negation detection, etc.) into a single API call, reducing the need for complex pipeline orchestration and local NLP implementations. The framework's modular design allows for seamless replacement of traditional NLP APIs with LLM APIs while maintaining its core benefits of flexibility and configurability.

chapter 4 employed various methods, such as rule-based parsing, named entity recognition, and HTML structure analysis, to extract ADR from SmPC documents. Although being an effective method with high precision and recall, it is quite limited in comparison with today's LLMs.

1. **Generalization:** The rule-based approach is tailored to SmPC documents and requires manual rule creation. LLMs can generalize accross document formats without explicit rules. By simply adjusting prompts, LLMs would process different product labels with ease.

2. **Context Understanding:** Traditional NER struggles with contextual understanding, while LLMs excel at understanding context and can better handle ambiguous cases.

3. **Error Handling:** To handle specific error cases (e.g., handling "or", "and" in ADR lists), manual checks were introduced. LLMs can better handle such variations naturally.

Traditional NLP techniques, such as topic extraction and text classification employed in other chapters can be easily replaced by LLMs. It simplifies the clinical text processing pipelines in this dissertation and improves the generalization of our approaches. However, due to the limitations of today's LLM models in explainability, transparency and reliability, extensive tests and evalation are necessary before deployment in clinial setting which requires trust, transparency and consistency. The architectural complexity and hundreds of billions parameters make it difficult to trace how LLMs generate outputs. Moreover, AI companies which built the best LLMs in the world are more reluctant to share how they train their models and what data is used. As a well-known feature of LLMs from the beginning, hallucination produces false information and incorrect responses which is a huge challenge in high-stakes domains like healthcare.

8.2.2 Future Work

This dissertation has contributed to overcoming challenges in healthcare information system engineering, especially in clinical settings. Besides developing frameworks and tools to solve specific HIS problems, we also introduced open source methodology as best practice for HIS engineering. Nevertheless, there are still many improvements and unexplored territories in this field. A few future research areas can be imagined, and we will describe three of them below.

1. **Transitioning from prototype to practice –** Due to limited resources and time, we only managed to validate the architecture proposed in chapter 2 via a multinational randomized clinical trial. Further research or experiments on applying the proposed tools and frameworks from Chapter 3-5 in practice are necessary to prove their true value to healthcare. The lightweight API-based NLP framework from chapter 3 could be integrated into hospital information systems (HIS) to automate clinical concept extraction from discharge summaries or medical records, thereby reducing manual effort and improving decision-making. Similarly, The big data framework for biomedical literature mining designed in chapter 5 could be deployed in clinical research institutions to analyze large-scale biomedical/clinical literature of any given topic. Moreover, chapter 4 constructed an adverse drug reaction (ADR) extraction pipeline which is

intended for the development of ADR knowldege base. Then the knowledge base can be used for automatic ADR detection from electronic patient records.

However, practical implementation requires addressing several challenges, including privacy and security concerns, computational efficiency, and seamless integration with existing clinical workflows. Therefore, before deploying anything in pratice we should start with a small proof of concept in a well-defined context.

2. **Leveraging LLMs in clinical text processing –** As discussed in subsection 8.2.1, LLMs are superior to traditional NLP techinques in many tasks. Future research should focus on integration of LLMs with frameworks or approaches introduced in this dissertation. Given the generalization of LLMs, it is worth investigating the replacement of various NLP APIs with LLMs, and get a more lightweight framework with better performance. As a follow-up research on ADR extraction pipeline from chapter 4, we could explore using LLMs to simplify the extraction pipeline and add multilingual support. chapter 5 only used two NLP tasks to validate the efficiency and effectiveness of the framework, more NLP tasks need to be tested. LLMs could make this very easy by just adjusting the prompts. Therefore, future research on integrating LLMs will improve the framework.

3. **Expanding knowledge sources –** As the largest code host platform in the world, Github hosted hundreds of millions projects with millions of active developers. However, besides GitHub, there are other platforms which also host open source projects, such as GitLab, Bitbucket and SourceForge. Therefore including open source projects from other platforms will provide a more hollistic view of open source clinical software. Likewise, literature search and collection in chapter 6 was limited to Google Scholar. Since Google Scholar often fails to index the latest literature in time, especially in fields with rapid development, like AI, we could fill the gap by including literature from arXiv. Another future research direction lies with developing a retrieval augmented generation (RAG) based chatbot on the expanded knowledge sources. The chatbot will offer an easier and interactive way for retrieving related literature for a given opensource clinical software.