KYOTO UNIVERSITY

A Computer-mediated Support for Writing Medical Notes with Coder's Perspective

by

Lukman Heryawan

Supervisor: Professor Tomohiro Kuroda

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Graduate School of Informatics
Department of Social Informatics

August 2020

Declaration of Authorship

I, LUKMAN HERYAWAN, declare that this thesis titled, 'A COMPUTER-MEDIATED SUPPORT FOR WRITING MEDICAL NOTES WITH CODER'S PERSPECTIVE' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Sometimes the best thing you can do is not think, not wonder, not imagine, not obsess. Just breathe and have faith that everything will work out for the best."

unknown

KYOTO UNIVERSITY

Abstract

Graduate School of Informatics
Department of Social Informatics

Doctor of Philosophy

by Lukman Heryawan
Supervisor: Professor Tomohiro Kuroda

Electronic medical records (EMR) are the primary point of data capture for patient care. EMR can also serve as data capture points for secondary usage such as clinical research and reimbursement management. Ideally, structured and standardized data can be captured in one go at the point of care and made available to meet the needs of payers, healthcare administrators, clinical research and public health.

For the primary and secondary usage of clinical data, medical coders (coders) need to transform human-created data, such as medical notes, commonly used by physicians to capture patients' data, into a structured and standardized format that machines can process. However, to accomplish that, some issues need to be addressed due to the way medical notes are recorded by physicians, particularly their input of unstructured data.

The first issue we found was that coders, who are separated from physicians, must assess the medical notes to assign diagnostic codes and medical procedure codes in adherence with International Classification of Diseases standards. Discrepancies between the physicians who write the medical notes, and the coders who assign the codes, may occur. In this study, such discrepancies were assessed by performing a video-based survey to understand the coders' perspective, informing the development of a writing support system to achieve unproblematic medical notes based on the coders' perspective. This survey found that problematic medical notes were not caused by a single problem but by multiple problems. Abbreviations including acronyms used by physicians were revealed to be the major problem in assigning diagnostic codes, whereas incomplete data were the major problem in determining medical procedure codes. This survey also showed that problematic SOAP notes may contain helpful keywords for coders that can help in determining diagnostic and medical procedure codes.

The second issue we found important was structured data recognition from medical notes. To parse free text medical notes into structured data such as disease names,

drugs, procedures, and other important medical information, first, it is necessary to detect structured medical entities or clinical terminologies from texts. It is important for EMR to have structured data with semantic interoperability to serve as a seamless communication platform whenever a patient migrates from one physician to another. However, in free text notes, medical entities are often expressed using informal abbreviations. An informal abbreviation is a non-standard or undetermined abbreviation, made in diverse writing styles, which may burden the semantic interoperability between EMR systems. Therefore, detection of informal abbreviations is required to tackle this issue. In this study, the Long Short-Term Memory (LSTM) based model was applied to detect informal abbreviations in free text medical notes with high precision by concatenating embedding, bag of words and word2vec vectors to the model. Additionally, sliding windows was used to tackle the limited data issue and sample generator was used for the imbalance class issue. This method was able to recognize informal abbreviations with high precision by using small data sets. The detection can be used to recognize informal abbreviations in real-time while the physician is writing and prompt the physician with appropriate terms for the informal abbreviation for confirmation, thus increasing the semantic interoperability.

The third issue we found important was about physician and coder interaction. Obtaining the data required by coders might be problematic due to a free text writing style or noisy writing style of physicians, such as use of informal abbreviations. It is difficult to handle the noisy writing style of physicians using a template-based approach and written text automation like auto-completion, therefore indirect guidance, such as reminders or an adviser-like user interface was proposed.

By considering the constraints of structured and standardized data, which are necessary for the medical coding task, together with the natural workflow including input of unstructured text, this doctoral research aimed to develop a new support system for writing medical notes with an agent representing the medical coders' viewpoint.

In this research, a video-based survey was used to elicit from coders their requirements. These were incorporated when building an LSTM based high precision detection model including an adviser-like user interface of a computer-mediated support system. The support enabled physicians to write medical notes more efficiently. The support also ensured that the medical notes contained more of the required structured data. The computer-mediated support is a human-to-human interaction-oriented support that is innovative in bridging human-to-human cooperation effectively.

This study can be applied to other domains which require structured data collection from free text writing, such as automation of general administrative documentation and bureaucratic work. For example, configuration of physician and coder cooperation is

akin to that often found in other fields, such as in the academic field where there is an author and a reviewer of a paper. The author is akin to the physician and the reviewer is akin to the coder. The computer-mediated support could support the author in writing the paper based on the reviewers' perspective. Thus, it can be generalized as a human-to-human interaction-oriented support for documentation based on a data requesters' perspective.

Acknowledgements

This doctoral thesis was able to be accomplished thanks to the advice and cooperation of various people.

First, I would like to express my deepest gratitude to Professor Tomohiro Kuroda who gave me the opportunity to enter this laboratory. Without his invaluable support, this doctoral thesis would not have been completed.

I would like to offer sincere thanks to Associate Professor Goshiro Yamamoto, Associate Professor Osamu Sugiyama, Doctor Purnomo Husnul Khotimah, and Assistant Professor Luciano Henrique de Oliveira Santos who inspired me with many discussions during my Ph.D. course. Without their teaching and mentoring on how to write an academic paper and presentation, I would not be able to publish journal papers and do better presentation.

I am very grateful to Professor Masatoshi Yoshikawa and Professor Hiroaki Ogata who give me invaluable advice of my doctoral thesis content. I am grateful to Associate Professor Hiromichi Mitamura and Professor Tadamasa Takemura who gave me much advice during my PhD progress. Also, I am thankful to Associate Professor Kazuya Okamoto and Assistant Professor Shusuke Hiragi for giving invaluable comments and suggestions.

I am also thankful to Professor Masayuki Nambu, Professor Hiroshi Tamura, Associate Professor Genta Kato, Associate Professor Shinji Kobayashi who gave me some advice in the research meeting and via email.

I would like to thank the Japan International Cooperation Agency (JICA), for providing me with generous scholarship and research funding for the duration of my research. Also, I thank to Design School program at Kyoto University, who provide me valuable knowledge and experience in design study and collaboration research, especially thank for Professor Tetsuo Sawaragi who gave me invaluable advice for my design-related research.

I would like to thank secretaries Yoko Hara, Yuko Furusawa, Kaori Shiomi and Kuniko Hiramatsu who assisted me with daily work, always gave me great support. Also, I am very grateful to all the Ph.D. course students and Master course students in the Medical Informatics laboratory.

I also send warm thank you to my colleagues in Universitas Gadjah Mada, Indonesia, who supported and collaborated with me to make all experiments and evaluations of this work possible. I am thankful to Doctor Agus Harjoko, Doctor Reza Pulungan, Doctor

Suprapto, Doctor Anny Kartika S, Doctor Lutfan Lazuardi, Doctor Nur Rokhman, Professor Triyono, Professor Chairil Anwar, and Professor Panut Mulyono.

Last but not least, I would like to thank my spiritual teacher, Doctor Aswin and my family who always offered me their love and support. To my father Professor Siswanto Masruri, mother Hartini, father in law Doctor Cholid Baidlowi, and mother in law Ulfah Ghozali in Indonesia, and my brothers, Ahmad Arief W and Muhammad Faris N, brothers in law, and sisters in law in Indonesia. Special thanks to my wife, Doctor Emi Azmi C and my two adorable kids, Aufa Hafidz A and Athaya Ilma R who always support me, thank you very much for your understanding and love. It is meaningful for inspiring me during hard time. Finally, we can pursue it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electronic medical records (EMR) are the central point of data capture for patient care. EMR can be used for both primary and secondary use. Primary use is for delivering health care to the individual from whom it was collected, such as in health checkups. Secondary use is for public health and clinical research. Digitization of EMR or other types of medical records and health information technology has expanded primary and secondary usage of clinical data [1]. EMR have great potential to improve the efficiency and quality of clinical research and reduce the cost of clinical trials. EMR systems can support both research and administrative related activities of clinical trials, public health and safety monitoring, and reimbursement management [2].

Ideally, structured and standardized data should be captured once at the point of care, and derivatives of this data made available to payers, healthcare administrators, clinical research and public health. The structured and standardized environment may significantly reduce clinicians' burden of administrating and capturing the data, dramatically reducing the time for clinical data to become available for public health emergencies and for traditional public health purposes. It should also significantly reduce the costs of distributing, duplicating or processing healthcare information, and greatly improve the quality of care and safety for all patients. In order to achieve that, EMR should be capable of transforming unstructured data, such as medical notes, into a structured format for both primary and secondary uses.

Ideally, medical coders (coders) should be able to transform human created data, such as SOAP notes commonly used by physicians for recording patients' diagnoses, into a structured and standardized format that machines can process.

In clinical research, data reuse will generate high-quality clinical evidence faster via better protocol feasibility assessment, improved patient identification and recruitment,

and more robust clinical study, including reporting serious adverse events. It will also maximize the value to customers and diversify revenue streams of research organizations, and enable the participation of clinical investigators and physicians in a larger number of clinical trials [3]. Clinical data reuse can also be of important commercial value [4]. Clinical data are used by public and private payers for cost-effective research and assistance with optimal reimbursement decisions; healthcare organizations store increasing quantities of clinical data for internal applications because of its potential as a very valuable asset. For the healthcare IT industry, research platforms allowing reuse of clinical data open new business opportunities facilitated by sustainable business models [5]. However, there are some issues that need to be addressed to reuse the EMR due to their recording nature, which is mainly using unstructured data entry like SOAP notes.

1.1 Problem

SOAP notes are the commonest input into medical records by physicians. These notes are used for assessing, diagnosing, and treating patients [6]. Each SOAP note is written in a structured and organized way, consisting of Subjective (S), Objective (O), Assessment (A) and Plan (P) parts. In the case of inpatient care, a discharge summary of the SOAP notes must be provided by the physician.

Coders, who work separately from physicians, assess SOAP notes to determine appropriate diagnosis codes, which are structured and standardized data, based on different SOAP parts. For example, in the case of reimbursement processing of insurance claims, in countries such as Indonesia, Thailand and Malaysia that use Diagnosis Related Group (DRG) for processing [7], information from the Assessment part are coded using the 10th Revision of International Classification of Diseases, (ICD-10) standard, while medical procedures from the Plan are coded using the Clinical Modification of the 9th Revision of the International Classification of Diseases (ICD-9CM) standard of the World Health Organization (WHO). The ICD standard defines standardized diagnosis and medical procedure terms and their corresponding codes, as well as the rules on how to assign them.

The DRG system groups hospital cases that are likely to have similar resource uses. It was introduced in America and adopted by many developed and developing countries worldwide. As DRG helps to control the cost of medical treatment, both hospital administrators and outside insurers like it. Hoverer, DRG may be used by some hospitals to adjust or manipulate data in seek of more profitable diagnosis and procedure codes for greater insurance or other reimbursements. They may even ignore some data that

brings no revenue to save on administration resources and costs, which is perhaps acceptable if the resultant coding mirrors the patients' actual conditions. Application of DRG would be suitable in cases where it fits the existing hospital infrastructure and human resources, whereas a potentially ideal coding process may be preferable. This would include two stages. First, upon patient discharge the physician compiles a standard discharge summary of the diagnosis and clinical treatments. Information about clinical care and medical records would be used to compile an accurate summary reflecting the patient conditions and clinical interventions. Second, a certified coder would assign appropriate ICD-10 and ICD-9-CM codes reflecting information in the discharge summary. In cases of ambiguity, the coder might examine the medical record and/or consult with the physician.

However, this would require a sufficient number of professional qualified coders, which may be impractical in some countries such as Indonesia, and also parity among hospitals to have access to these coders. As is often the case in other health professions, most qualified coders work in large urban area hospitals. Hospitals lacking qualified coders might have to send office staff to training workshops so they can function as part-time quasi coders.

Hospital income and reimbursement of medical costs relies on accurate records of what work was performed. Accordingly, patients' diagnoses, test results, and treatments have to be documented for reimbursement and also to ensure quality care and good practice, especially in chronic cases. Each patient's health information should be available for reference in event of subsequent complaints and treatment requirements. And that information must be unambiguous for multiple doctors and health care staff to understand.

One difficulty to achieve this, is that there are countless numbers of conditions, diseases, injuries, causes of death, and a constant growth in novel diseases and treatments such as the new corona virus COVID-19. Thousands of services are performed by providers requiring a multitude of treatments, drugs, and supplies to be tracked. Medical coding classifies these to facilitate reporting and tracking. However, there are multiple descriptions, abbreviations, acronyms, names, and eponyms for diseases, procedures, and tools. Medical coding standardizes these to render them unambiguous and more easily understood, tracked, and modified.

Coding is straightforward in many cases. With experience, coders develop an understanding of the procedures and practices of their employers' facility. They may occasionally encounter a medical note that requires more of their time to perform in-depth research to code correctly. Even commonly used codes may contain gray areas for inexperienced coders. In complex or unusual cases, coding guidelines may be difficult

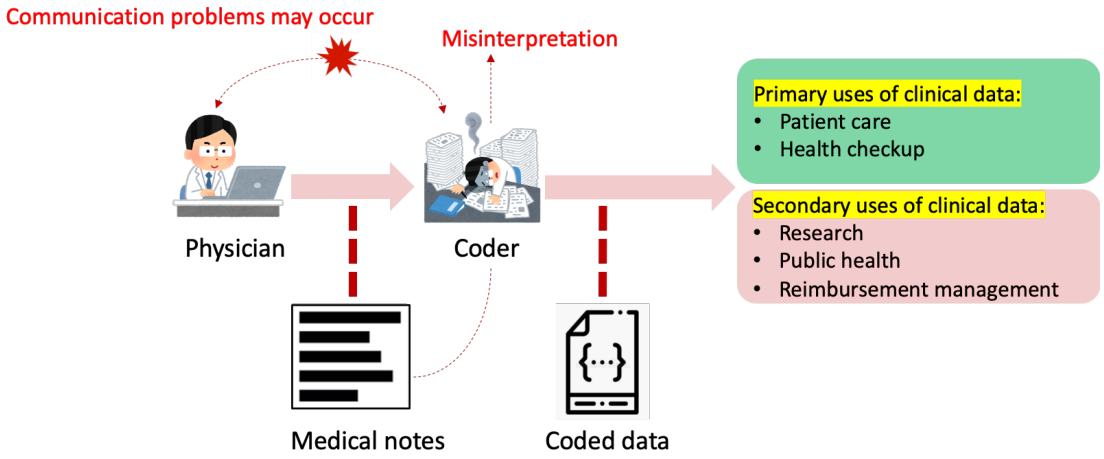

FIGURE 1.1: Problem in the medical coding task

to interpret. Experienced coders may consult inhouse with peers and professionals or discuss nuances by communicating online with specialists they have met at national conferences, etc. Inservice training and coding-related periodicals such as Healthcare Business Monthly also provide opportunities for coders to develop experience, and advance the understanding and professionalism of coders.

An issue that frequently recurs during the medical coding task is the coders' struggle to assess SOAP notes written by physicians. When necessary, coders should consult a physician for clarification of the contents and confirm whether the coders' interpretation is correct. However, due to the coders' and physicians' heavy workloads, arranging a consultation may be hard. Moreover, miscommunication between the physician and coder might occur during the consultation. Figure 1.1 illustrates the issue.

A typical approach to tackle this issue is to use a structured data entry system (SDES) or a template-based approach. In SDES, data entry has predefined categories and conditions. As well as recording patient history or physical examination findings, SDES can be used to develop specific flow sheets tailored to a certain disease process or procedure. SDES allow for uniformity and standardization of data entry and collection, which in turn facilitate reporting data and using template-based notes for EMR [8].

Developing and utilizing SDES is critical for both optimizing delivery of top-quality patient care and collecting data for primary and secondary use. SDES are critical because most clinical terminologies or concepts are only available in EMR as free text, an unstructured form. The most frequently available concepts are demographic elements, captured by all hospitals, followed by diagnostic and procedural entities. Other data elements are captured far less often. A properly designed SDES can boost patient-oriented research, and facilitate medical advances, as well as the completeness and accuracy of the EMR data [9]. A previous study [10] proposed four steps to create a usable SDES:

1. Institute a clinical advisory committee to develop and maintain standards for clinical protocols for clinical information within EMR,
2. Identify the "deal breakers" for structured data entry, especially in regards to physician resistance,
3. Identify the workflows to facilitate data entry capture, and
4. Identify the technology platforms necessary for seamless integration, often with the help of information technology or services departments.

General descriptions in medical records, such as progress notes, examination reports, operation reports and summaries, are so diverse and complicated that these data are generally entered as free text in EMR. In order to use these data for research, clinical evaluation and so on, one possible method is natural language processing [11]. However, to achieve perfect results by this method, all of the words in the entire medical field, including abbreviations (henceforth including acronyms) and frequent typing errors, have to be entered into this system beforehand. This would entail a tremendous amount of work. The strategy of template-based data entry is a practical method from the viewpoint of data analysis [12] without considering machine learning.

1.2 Template-based systems

The Oxford Dictionary defines "template" as "something that serves as a model for others to copy." Templates are very often used in medical practices to reduce the workload associated with having to collect the same information repeatedly. Without them, it can be time consuming for clinicians to collect and record clinical information during each patient encounter, whether making records on paper or directly into an EMR. However, in order to use templates effectively and avoid risky practices, several important elements must be considered.

Firstly, there are two types of EMR templates, static and dynamic. A static template contains empty data fields where a user simply fills in the blanks or selects items from a list or drop-down menu. Each time the template is used, the data fields need to be completed from scratch. A dynamic template works similarly, but automatically populates certain data fields from information that is collected by and stored in the EMR. For example, when opened, the template may automatically populate certain fields such as the most recent laboratory results or vital signs collected by a nurse prior to the doctor-patient consultation. These dynamic or "smart templates" can be designed for a variety of complex conditions to ensure that all the required information is

recorded in a standardized and repeatable format. They are ideal for managing patients with chronic diseases such as diabetes mellitus or chronic kidney disease [13].

Another consideration is that there are several ways to create templates. A more traditional approach is to define which data is to be collected via the EMR and then use a set of tools to lay out the data fields before saving as a template. Some EMR allow users to create templates "on-the-fly." This means that as the physician uses the EMR during the clinical encounter, SOAP notes, medical treatments and default values can simply be saved at the end of the encounter as a template for a specific examination or condition. Standardized data are required when creating templates. If the EMR allows physicians to define drug names or add specific codes for diagnoses, medications or other clinical data, it can be extremely difficult in the future to find data for reporting purposes, a problem that can be further amplified if physicians use templates created by another practice that has a different customized naming structure. Templates can be created with large narrative text boxes, which may be desirable for certain specialties, e.g. psychiatry. However, data recorded in narrative format cannot be easily searched and analyzed [14].

Template-based EMR have been identified as a significant contributor to physician burnout, emotional fatigue and job dissatisfaction [15]. It is also a barrier to accepting EMR technology, according to the Unified Theory of Acceptance and Use of Technology (UTAUT) model [16]. Co-writing medical records, such as SOAP notes, might solve these problems. For example, the physician may be able to reduce their workload by inviting their patients to update or input their symptoms into the medical records before their visit.

1.3 Co-writing of medical notes

Social cooperation among healthcare professionals is necessary for smooth patient care and hospital operation [17]. A physician-patient team-based approach is essential for good quality patient care. Recently, a system of physician and patient co-writing medical records has been implemented, which found that patients may benefit from co-writing medical notes [18]. Co-writing may also reduce the time physicians need to spend on making medical records. It also makes for a more patient-centered approach and makes sure that the patient's voice is heard.

In parallel with the coder-centered approach that requires structured and standardized medical records for optimal usage of clinical data, co-writing of medical records between

physician and coders can be achieved as long as there are enough coders to implement a one-to-one co-writing relationship with the physicians.

Most of the writing collaborations are asynchronous [19]. The writing tools should have specific features to support collaboration, such as the maintenance of a revision history and some indications of which changes were made in the revision, such as Microsoft Word's "track changes" function and a choice of how much of the edits one wants to see. For example, Word allows users to see the original version with changes marked or the final version without changes shown, so one could review the document for readability. But today, there are new tools and features available to support synchronous collaborative writing, and they are being used more commonly. Recent research on people's use of Google Docs revealed that when one makes changes to another's writing, it creates a social dynamic [20]. Co-authors who are not mutually familiar are more likely to comment on the text rather than make changes directly. One study [21] found that, in general, people think the perceived quality is higher when the writing is seen by others. On the other hand, some felt that having someone else directly edit their texts was intrusive, and lowered their sense of ownership.

1.4 Approach

A common way to optimize the medical coding task is to develop a template-based system. However, template-based systems often require many fields and take too much of the physician's time to enter all the medical data. Another approach is co-writing, whereby physicians and coders document SOAP or other notes together. However, that approach seems impractical, with lack of coders preventing a one-to-one coder-physician co-writing relationship.

This doctoral research aims to develop a computer-mediated writing support system for physicians which represents the medical coders' viewpoint. The system mimics a coders' interaction with a physician to obtain unambiguous structured data. Consideration was given to the constraints of structured data required by a coder and unstructured text that may be input by a physician documenting medical notes. This computer-mediated support system models a physician-coder interaction informed by the coders' perspective.

In this research, a video-based survey revealed what problems coders are faced with when trying to code medical notes and what requirements they need as a result. An LSTM based model was developed that was capable of detecting the problems with high precision and has an adviser-like user interface (UI) that suggests remedies to support both physicians and medical coders in the medical coding task. The writing

support enables physicians to write more efficiently and coders to capture more required structured data. In addition, it also promotes human-to-human interaction.

An innovative approach is introduced to support the writing of medical notes based on the perspective of a data requester such as a coder. It enables physicians to write medical notes without hampering their natural flow of writing in the EMR system.

1.5 Contribution

The main contributions of this doctoral thesis are the following:

1. Proposal from the coders' perspective, a support for writing medical notes.
2. Development and evaluation of a structured data detection model, using abbreviations as a case study.
3. Confirmation of the usefulness of the proposed writing support system.

This doctoral thesis clarifies the issues, and organizes the requirements, in the development of a computer-mediated writing support system from the coders' perspective. As for the first contribution, a video-based survey was conducted to ascertain coders' perspectives when dealing with problematic SOAP notes. The survey was used to set the system's features and confirm that multi-detection features are required to accommodate the coders' perspective in supporting the physician's writing.

As for the second contribution, an informal abbreviations detection model using deep learning, one of the system's main features, was developed. Its ability to recognize informal abbreviations from free text medical records and precisely discriminate between abbreviations and non-abbreviations was confirmed by precision evaluation targeting a public English medical notes data set from www.mtsamples.com.

As for the third contribution, the usefulness of the proposed writing support system's UI was evaluated by several physicians. It was confirmed that the proposed UI is useful in supporting physicians to write SOAP notes, and ensure that the SOAP notes were not problematic from the coders' perspective. These contributions can promote future research on the clarification of computer-mediated human-to-human cooperation, which promote better interaction between physicians and coders in the medical coding task. This work may also support the exploration of new medical notes entry strategies based on the above findings, and even in the development of more usable applications to establish and sustain better transformation of human-originated data to machine-readable data.

1.6 Thesis structure

Chapter 2 gives an overview of work related to this research, specifically in the field of support for writing medical notes. The overview includes existing approaches, such as auto completion and copy and paste, as well as support for writing medical notes in several countries. The current status of research is also briefly described.

Chapter 3 illustrates the background of the proposed methodology, together with its main concepts and requirements.

Chapter 4 describes the results of a video-based survey designed to understand the coders' perspective in dealing with problematic SOAP notes, enabling the design of a support for a medical notes writing system for physicians.

Chapter 5 describes deep learning modelling using LSTM to detect informal abbreviations among sentences in medical notes, which is a vital task for structured entity identification in EMR. Additionally, our approach expands the field of application of the LSTM model since the technique has not been used hitherto for this kind of task.

Chapter 6 describes our proposed system's user interface and gives an appraisal of its differences with common medical record user interfaces in terms of its efficiency in writing medical notes and its usefulness in capturing more data required by the coder. It includes the results of a preliminary quantitative evaluation comparing the proposed user interface and the common standard medical record user interfaces.

Chapter 7 discusses the main findings of this doctoral research, as well as describing its current usability and limitations.

Chapter 8 concludes the doctoral thesis by summarizing the main contributions of this doctoral research, as well as their impact on future research.

Chapter 2

Related Work

This chapter is an overview of work related to this research, specifically in the field of support for writing medical notes. It includes existing approaches for support for writing medical notes, such as auto-completion and copy and paste, as well as support for writing medical notes in several countries.

Physicians write notes for a variety of purposes, including documentation of care, communication with other team members, and providing evidence of services for billing. Ideally and preferably, physicians would have an assistant or co-writer to write medical notes efficiently and to produce coded data. However, since co-writing between physicians and coders is infeasible, physicians require computer-based writing support systems for efficiency and to be able to produce coded data if necessary.

There is a growing number of applications that help physicians write notes efficiently and to produce coded data. A key question is for whom physicians are writing notes (coders, themselves, patients, or other physicians), and many notes written by medical professionals go largely uninterpreted by others [22].

Today's medical records are not only a clinical record intended for informing other physicians and healthcare workers of clinical findings, thoughts and plans, they also provide clinical data for secondary usage, such as invoices, the basis of quality assurance and improvement by hospital review committees, and for data collection in support of research [23].

Some EMR systems include computerized order entry for physicians and decision-support tools for improving patient care. They can include clinical data for primary and secondary use. While these features can save physicians time by providing easy access to

masses of information, they do not necessarily save time in the day-to-day entry of patient encounter notes or SOAP notes [24]. Therefore, it is understandable why physicians want ways to enter information into the EMR more efficiently.

Copy-pasting information within EMR is one of the time-saving tactics seen in many medical centers [25]. This takes many forms: large sections of notes can be copied without making changes from one day to the next during a hospital stay; new information can be added to copied information to create running notes; a provider can copy part of notes from another provider; and notes can be copied from one patient to the next.

Despite its widespread use, copying and pasting notes raises a number of ethical issues. It has been shown to produce notes that are confusing, increasingly lengthy, uninformative, disorganized, internally inconsistent, misleading or lacking in credibility, and may introduce and propagate errors that can place patients at risk and may be unsuitable for proper primary and secondary usage of clinical data [26].

In another approach to save time, physicians dictate reports of daily progress within the hospital. While this may address some of the copy and paste issues, this approach is less common because transcription introduces significant delays and costs [27].

Automatic speech recognition software (ASR) is increasingly used by physicians. Using ASR, clinical documentation would be an automated process, with minimal necessary input from humans. For this automation, a digital scribe that has an automated clinical documentation system capable of capturing the clinician–patient conversation and then generating the documentation for the encounter, as a human medical scribe would do, has been developed [28].

However, in regard to extraction of structured information from medical notes, large-scale semantic taxonomies such as the Unified Medical Language System (UMLS) was designed for written text, not for spoken medical conversations or spoken text [29]. Therefore, differences in spoken and written discourse can cause inaccuracies and word mismatching when existing tools are used for medical conversations. Consequently, additional steps must be taken to identify semantic types and groups to control the way spoken discourse is mapped to medical concepts, which can be a time-consuming trial and error process [30].

Digital automation of written text, such as auto-completion and auto-correction that benefit from language models [31], can be used as a proper alternative for a physicians' writing support system that is also more accurate for extracting structured data from medical notes. The models can effectively predict a lot of the content of physicians' notes if the maximum context provided by the EMR is sufficient. In many cases, however, the maximum context provided by the EMR is insufficient to fully predict the notes [32].

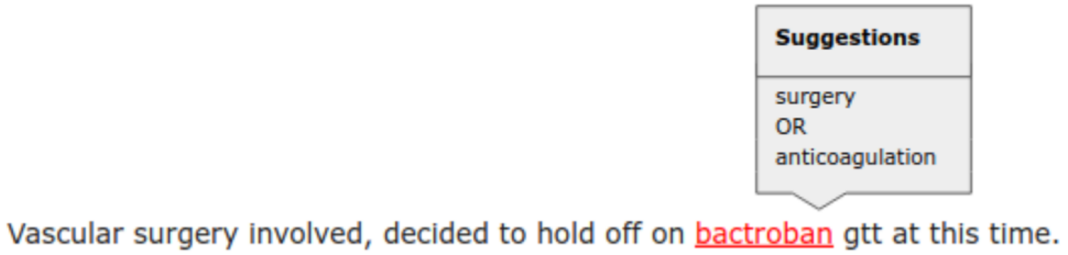

FIGURE 2.1: Auto-correction support to write medical notes [32]

Therefore, the cost of a mistake in auto-completion and auto-correction is potentially high. Because of this, auto-completion and auto-correction systems must have high accuracy to be of use [33]. Figure 2.1 shows a previous study's auto-correction support for writing medical notes [32].

Modification of templates or SDES modification in some countries including the USA and Japan is expected to improve the usability of the EMR system. In the USA, writing medical notes is assisted by structured narratives [34]. A structured narrative is where unstructured text and coded data are fused into a single representation, combining the advantages of both. Each major clinical event (e.g., encounter or procedure) is represented as a document to which Extensible Markup Language (XML) is added to indicate gross structure (sections, fields, paragraphs, lists) as well as fine structure within sentences (concepts, modifiers, relationships). This form of representation is semi-structured in that the gross structure imposes restrictions on the clinician (standard fields for data entry), while allowing freedom of expression within those units (free text paragraphs). All of these structured elements are associated with standardized codes that enable them to be reused for various computational purposes. Figure 2.2 shows a structured narrative user interface for writing medical notes in the USA [34].

In Japan, physicians commonly enter clinical data directly into an EMR system. This requires fast input handling. In addition to clinical care, the entered data should provide the basis for synthetic and analytic purposes in clinical research, education and hospital management. To merge the operability and the utility of an EMR system, a system called DTDES ("dynamic template" driven data entry system) has been developed to allow users to select the desired data from a list of items by means of a graphical user interface (GUI) [35]. Figure 2.3 shows a dynamic template user interface for inputting medical records in Japan [35].

In Indonesia, medical notes are still being created in free text format, such as SOAP notes, without a writing support system. Most of the medical records in Indonesia are

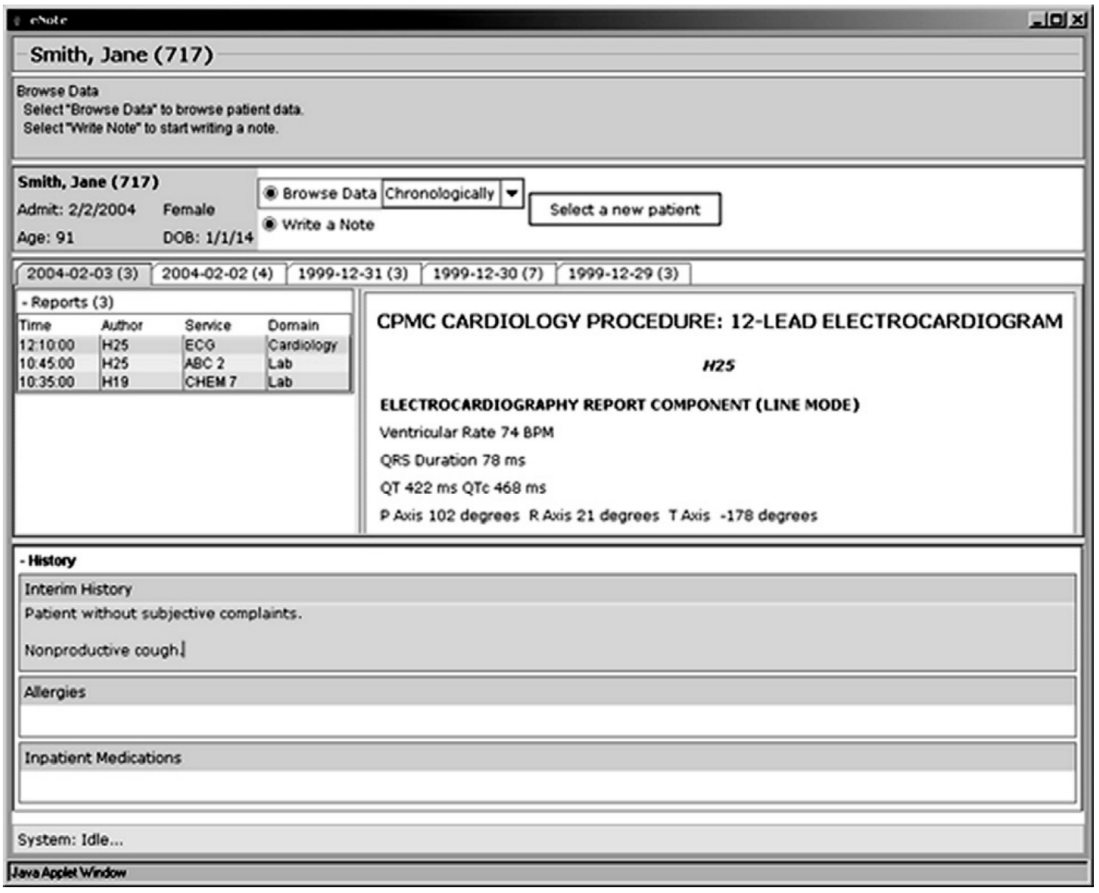

FIGURE 2.2: Structured narrative user interface [34]

unstructured data [36]. Therefore, for secondary usage of clinical data, such as insurance claim reimbursement, Indonesian hospitals have to rely heavily on coders' availability and expertise in deciphering medical notes into structured data.

Figure 2.4 illustrates the medical coding procedure used in Indonesia for reimbursement of insurance claims. Coders assign codes by assessing the SOAP notes and, if required, other medical records including inpatient discharge summaries. The assigned codes are input into Diagnosis Related Group (DRG) software, which groups patients with similar hospital resource utilization and similar clinical characteristics [37]. The DRG system requires diagnostic codes and medical procedure codes to group patients using a United Nations University (UNU) grouper and to determine reimbursement for insurance claims. The UNU grouper is a universal software grouper using ICD-10 for diagnosis classification, and ICD-9CM for medical procedure classification [38]. The last step in insurance claim reimbursement is for the internal verifier in the hospital to validate the codes and verify the insurance claim i.e. to determine whether the codes accurately represent the contents of the SOAP notes and the amount being requested.

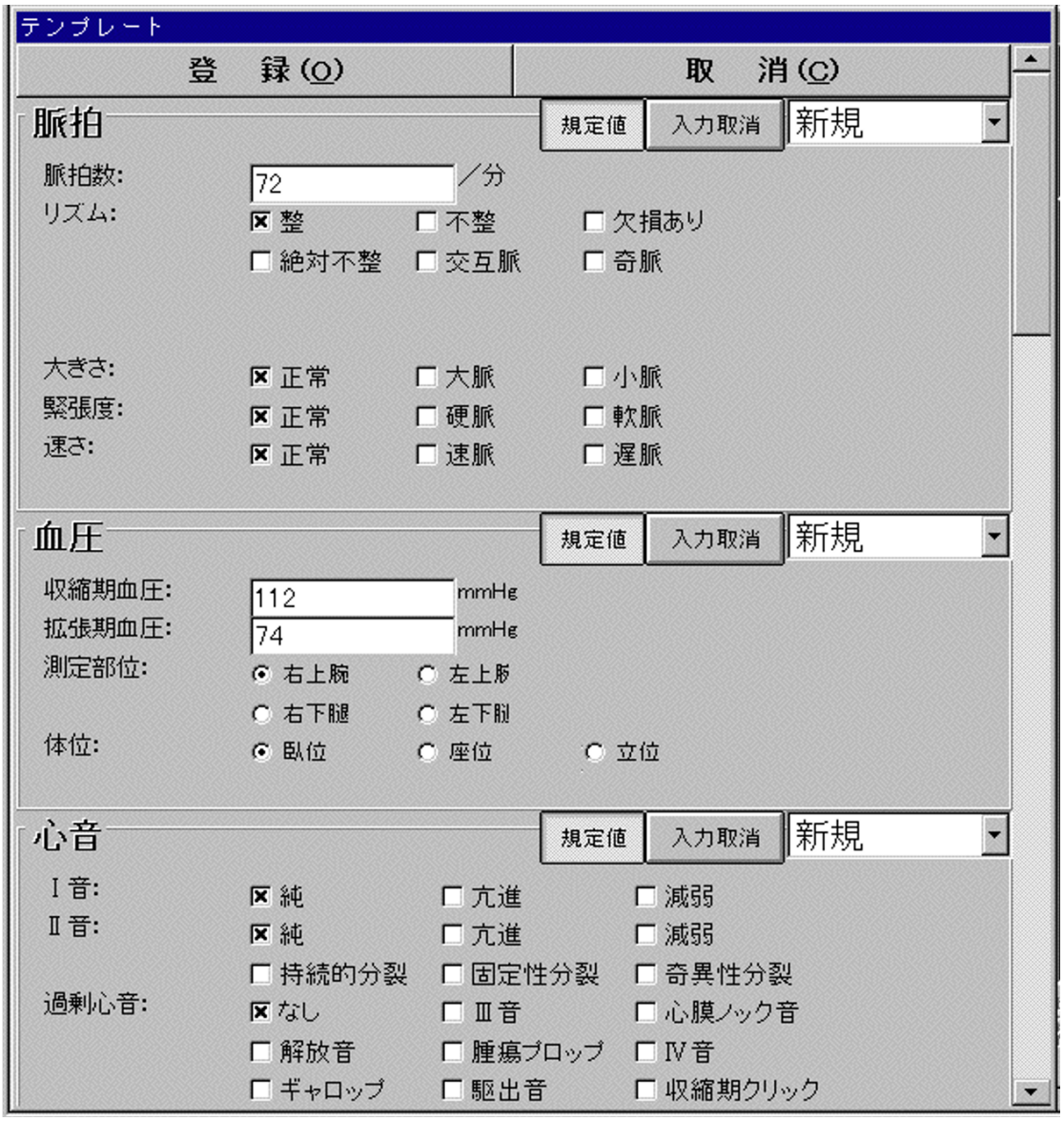

FIGURE 2.3: Dynamic template user interface
[35]

After verification, the final insurance claim document is sent to the National Health Insurance (NHI) agency for external verification and reimbursement.

2.1 Position of research

Cultural barriers to adopting standards and the use of structured data do exist. While structured data is required to aggregate, report and transmit the collection of data at the point of care, physicians often perceive it as hindering their ability to practise medicine and make notes efficiently. Adoption of EMR systems that allow physicians flexibility in making notes by free text recognition technology has flourished. Physician productivity

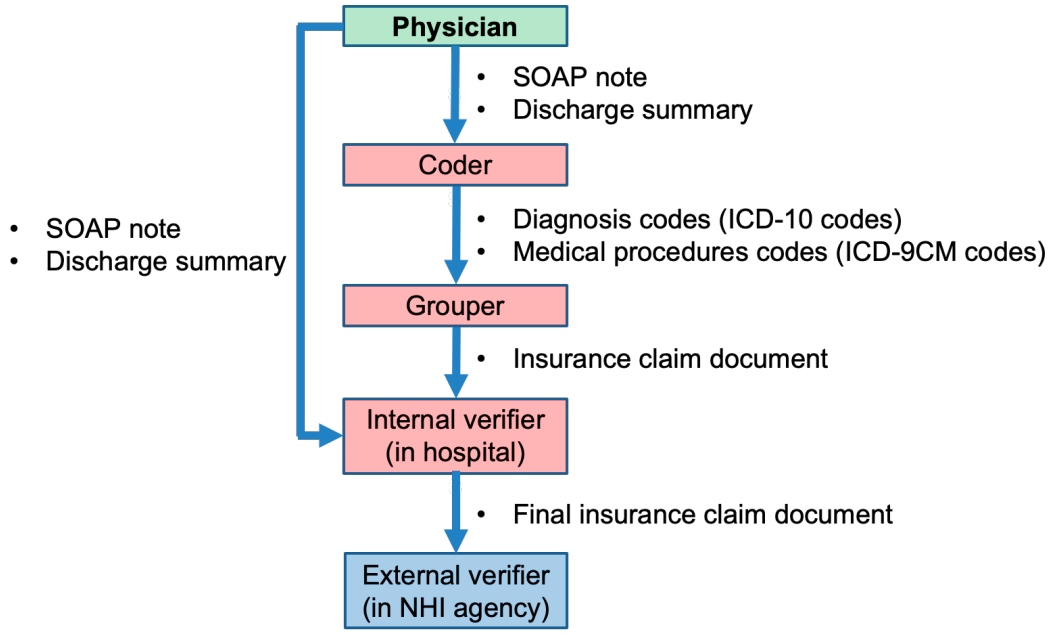

FIGURE 2.4: Flow of medical coding tasks flow for reimbursement of insurance claims in Indonesia

decreased due to frustration over EMR burdening the capture of structured clinical data at the point of care, immediately following a patient visit, or at the end of the day.

To support flexible documentation in the form of physicians' written free texts and to extract structured data to help coders tasked with medical coding, a support for writing medical notes from the coders' perspective has been proposed. It is innovative in that the natural way physicians write is maintained and the coders' perspective is effected as autonomous coder representatives or agents. This has never been reported in previous studies. The approach mimics the natural physician-coder cooperation context, whereby physicians write medical notes in a natural manner while guided by the coders' suggestions. These suggestions are given from the coders' perspective, that is, so that the physician will make medical notes that do not contain problems for coders. An agent or autonomous representative performs their task in the physician-coder cooperation, such as assessment of the structured data and data relating to inquiries. Figure 2.5 shows our research compared to other recent approaches regarding a support system for writing medical notes.

Regarding the physician-oriented support, there are two main approaches: transcription of dictations and automatic speech recognition. Ideally, these should allow the physicians to use their natural expressions. Regarding the coder-oriented support, the intention is more to increase the data required by coders or other parties, which need structured data for the primary and secondary usage of clinical data.

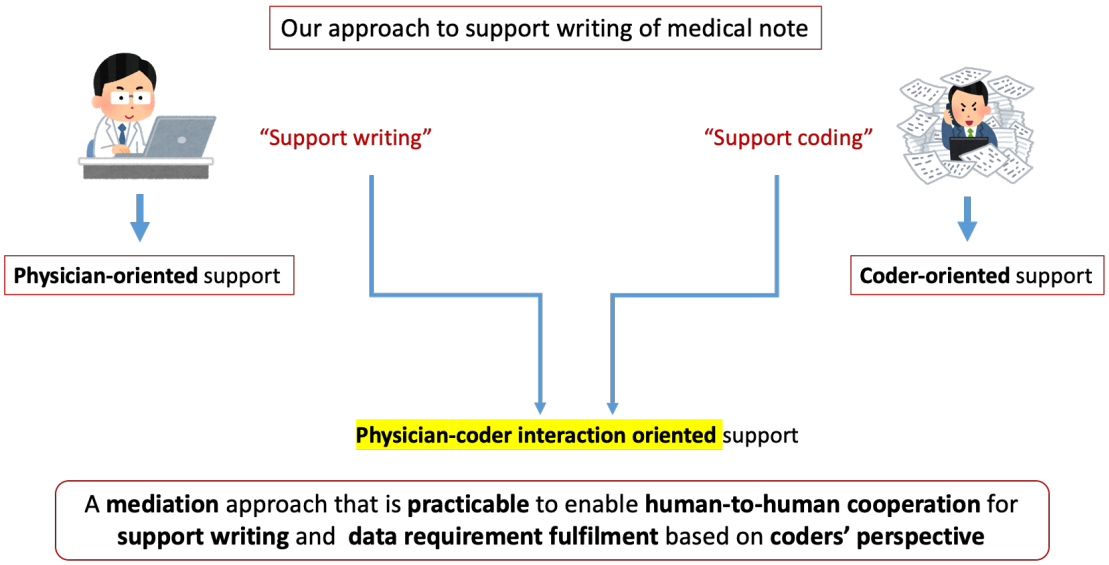

FIGURE 2.5: Position of research

The most common physician support is the copy and paste feature. It is found in most word processing applications, and therefore familiar to physicians. It increases their writing efficiency, especially when there are repetitive expressions. Although it can be used to increase the amount of data required by coders, it may produce a lack of data or inconsistent data not required by the coders. Overuse of copy and paste can therefore be a burden to coders, while aiding the physicians. Coder-oriented support can include dynamic templates adopted in Japan, structured narratives adopted in the USA, and written text automation such as auto-completion and auto-correction found in most virtual keyboard applications today.

The support system limits the physician's control with respect to dynamic templates, structured narratives, and written text automation, in order to guide physicians in providing data required by coders. However, if there are restrictions not based on the natural interaction between physicians and coders, one of the parties could be burdened. For example, when written text automation is not making 100% accurate text predictions because the natural way of writing medical notes is not being supported, the data required by coders might not be obtained as intended. In addition, controllable support should be developed by relying on the controller's perspective or data requester's perspective, which for the medical coding task is the coders' perspective. Relying on the requester's perspective may preserve the natural interaction between the physician and support system, thereby mimicking real interaction between the physician and coder in the support system.

In order to have full control over how physicians write medical notes while guiding them to write in a natural manner, a novel writing support system based on the coders'

perspective is desired. Incorporating the requester's perspective, i.e. the coders' perspective, in a support system that allows physicians full control of the way they write medical notes is an innovative contribution to the field of computer mediated human-to-human cooperation.

Chapter 3

Approach

This chapter gives a background to the proposed method and describes the main concepts and requirements. Three design considerations are required to realize computer-mediated support for writing medical notes from the coders' perspective: features, recognition, and the interface.

A common way to optimize medical coding is by structured data entry or template-based SOAP notes [39] as shown in figure 3.1. However, a template may contain many fields requiring much time for physicians to complete. In addition, physicians may be more familiar with free text-based notes (Appendix A) as mentioned previously [5]. Another approach is co-writing by physicians and coders to document SOAP notes. However, that is impractical as there are not enough coders. Therefore, one possible approach is to maintain free text-based SOAP notes and develop a support system that includes the coders' perspective to achieve non-problematic quality.

Several countries, such as Indonesia, Malaysia, Thailand, and USA utilize a "dual organization" model commonly seen in traditional hospital organizations. In this model, coders are a separate entity or person that codes the physician's SOAP notes. In non "dual organization" models, physicians do the medical coding by themselves. This model is adopted in Japan. There are more medical-coding task issues in "dual-organization" models. In Indonesia in particularly, physicians' SOAP notes may be misinterpreted by coders. To bridge the interpretation gap, a co-writing approach could be used. One approach is human-to-human co-writing.

In human-to-human co-writing, one possibility is for physicians and coders to write SOAP notes together in a cooperative manner. Thus, during every patient's check-up, a coder should accompany the physician. Professional communication and team collaboration is a hallmark of health care systems [41], as cooperation among healthcare

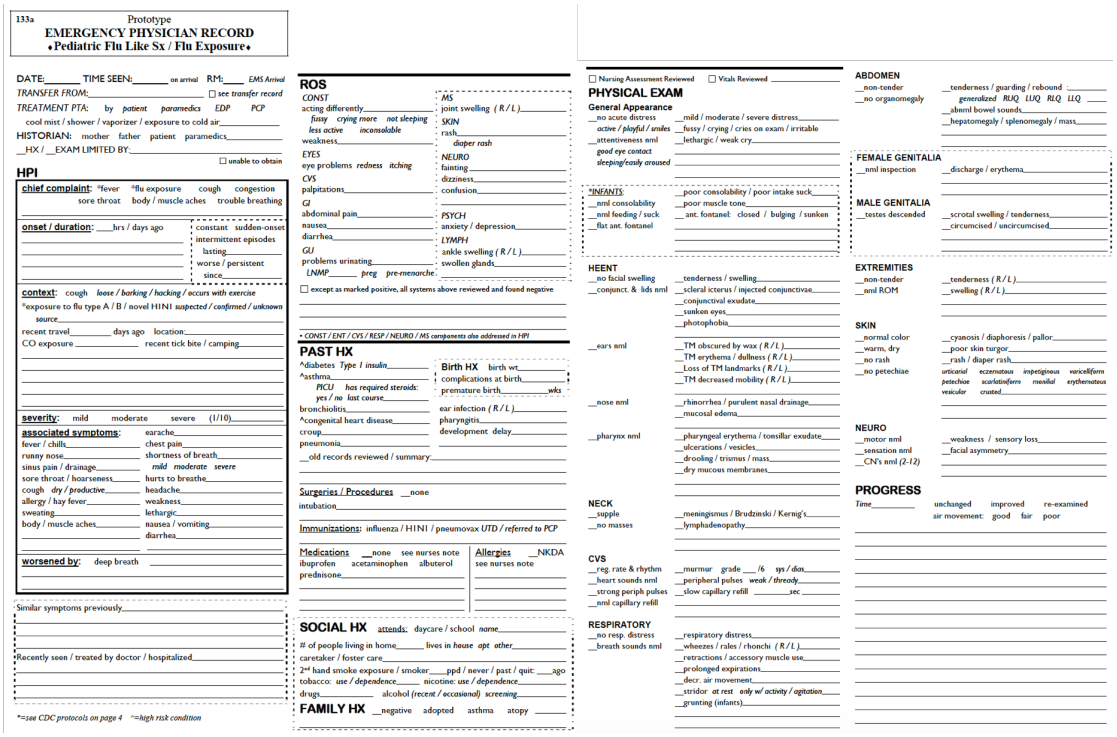

FIGURE 3.1: Example of template-based SOAP [40]

professionals is necessary to optimize patient care and hospital operation. This approach, however, is limited by the lack of human resources and competence. It may also infringe on privacy and be economically infeasible.

Rather than assigning one coder to each physician, human-machine/agent co-writing that uses a computer agent system that represents the coders' perspective has been adopted, resulting in virtual coder-physician cooperation in writing SOAP notes. This agent is an autonomous representative with its own purposes that senses its environment and with which people can interact [42]. Figure 3.2 shows the hierarchical categorization of such an agent [43]. Several autonomous agents have been developed in medicine, including ones that support the delivery of mental health therapies and others that incorporate automated responses into text-based tele-consultation systems. To date, most agent-based systems have focused on physician-patient interactions [44]. In addition, agent-based systems may resolve some of the physical limitations of a human representative. For example, agents may be duplicated to be more available than human representatives.

In proposed writing support systems, the agent makes suggestions to the physicians to standardize the SOAP notes and structure them in accordance with the coders' perspective. Figure 3.3 shows the concept of our proposed design of a writing support system with an agent representing the medical coders' viewpoint.

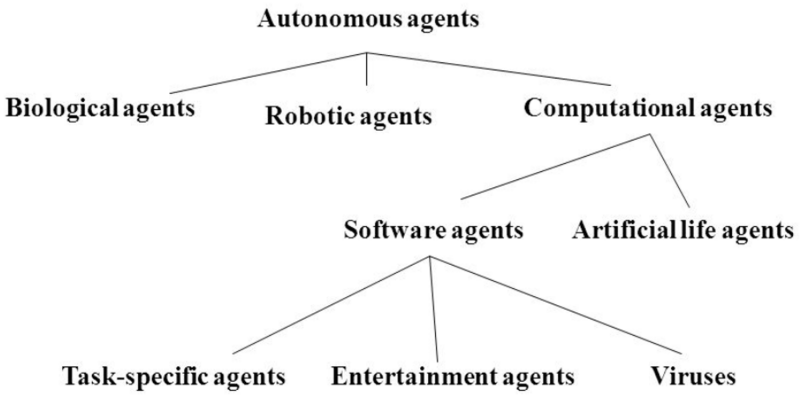

FIGURE 3.2: Hierarchical categorization of agent [43]

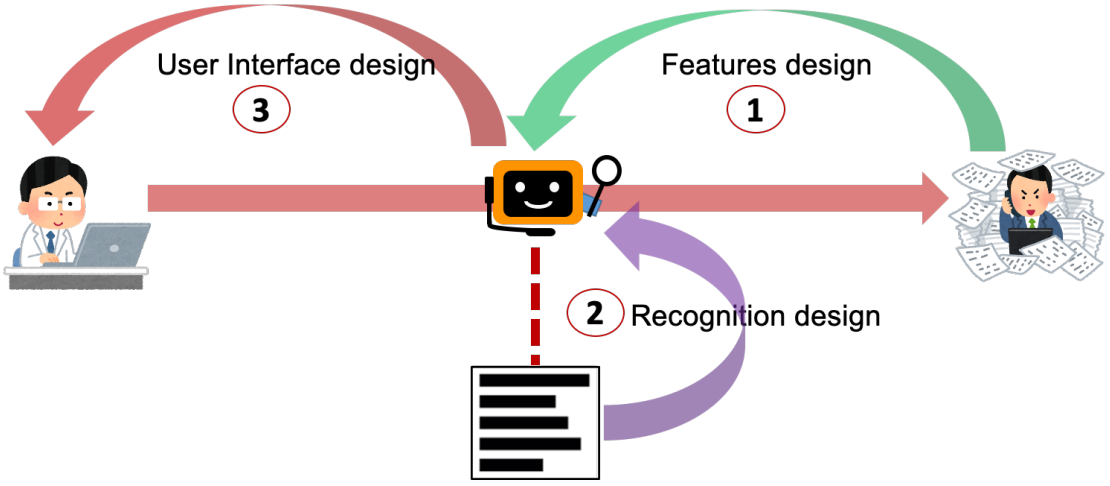

FIGURE 3.3: Design concept of proposed system

To develop the support system, it is necessary to determine the coders' perspective when viewing problematic SOAP notes and assigning an ICD code, which is dependent on experience. After determining the coders' perspective, the system features can be defined.

Secondly, coders' perspective label extraction from SOAP notes is conducted. The development and evaluation of the middleware or engine of the system's detection is per formed.

Thirdly, interaction between the system and the physician writing SOAP notes is observed, and the physician's experience is appraised.

Chapter 4

Features Design of a Support System for Writing Medical Notes

This chapter presents results of a survey designed to understand the coders' perspective in dealing with problematic SOAP notes and form the basis of the design of a medical note writing support system for physicians. This chapter was published in ABE (Appendix F).

4.1 Overview

SOAP notes are commonly used in medical coding tasks for reimbursement of insurance claims [45]. In Indonesia as our case study, medical coders, who are independent of physicians, evaluate physician-written SOAP notes to produce diagnostic codes based on ICD standards. ICD standards contain standardized diagnostic and medical procedure terms and their corresponding codes, as well as rules by which codes are assigned [46]. One problem during medical coding is the inability of coders to decipher SOAP notes written by physicians. Coders who find descriptions in SOAP notes to be problematic should speak with the physicians to resolve any problems [46]. Coders must confirm any problematic parts of SOAP notes to calculate reimbursement for insurance claims, and physicians must rewrite these notes based on their discussions. These revisions of SOAP notes are quite frustrating and burdensome for both physicians and coders, indicating a need for computer systems to provide writing support for medical coding tasks.

The present study describes the results of a survey to determine the coders' perspective in dealing with problematic SOAP notes, enabling the design of a SOAP note writing support system for physicians. Coders evaluated SOAP note typing videos, mimicking

the physician's typing behavior. The coders assessed whether or not the SOAP note typing video was problematic and provided comments on problems encountered on the SOAP note typing video. Video-based surveys are commonly used in the medical field. For example, some studies adopted this approach to collect data and analyze the survey contents [47, 48]. Our video-based survey is different from text-based surveys, as the latter present static text or survey questionnaires in the form of text to the subjects. In our case, a video-based survey is a dynamic form of a static or text-based survey. To make the videos for the video-based survey, the scripts of existing SOAP notes were typed and recorded on video to simulate being produced by the original physicians. The simulation is our video-based survey tool. By using a video-based survey, it is possible to acquire richer data than with text-based surveys. For example, not only comments, but also the time sequence of the comments can be acquired. To analyze the data, subjects labeled the location of their comments in the video. The quality of the acquired data was better with the video-based survey than with a text-based survey [49].

This study contributes to understanding the coders' perspective, which can be extended to develop the features of the system. Although several previous studies assessed the coders' perspective [50, 51], those studies focused on improving the coders' expertise [50] and the physician's attitude toward [51] medical coding tasks. In contrast, the present study proposes the development of system features that optimize medical coding tasks. To our knowledge, this is the first study designed to understand the coders' perspective in developing features of the system.

4.2 Observation to reveal coders' perspective

To understand the coders' perspective, an experiment was performed, in which physician writing of SOAP notes was modeled doing a medical coding task. In this study, dichotomous responses and free text responses were collected from the video-based survey. The content of the free text responses and comments were processed to yield categorized comments, which were further evaluated to yield spatialized comments corresponding to coder labeling of the comment location. Spatialized comments were designated categories and mapped to their position or location in the SOAP notes typing video screen. The dichotomous responses, including categorized and spatialized comments, were labeled "problematic" and "non-problematic" based on the coders' perspective. Figure 4.1 shows our video-based survey design.

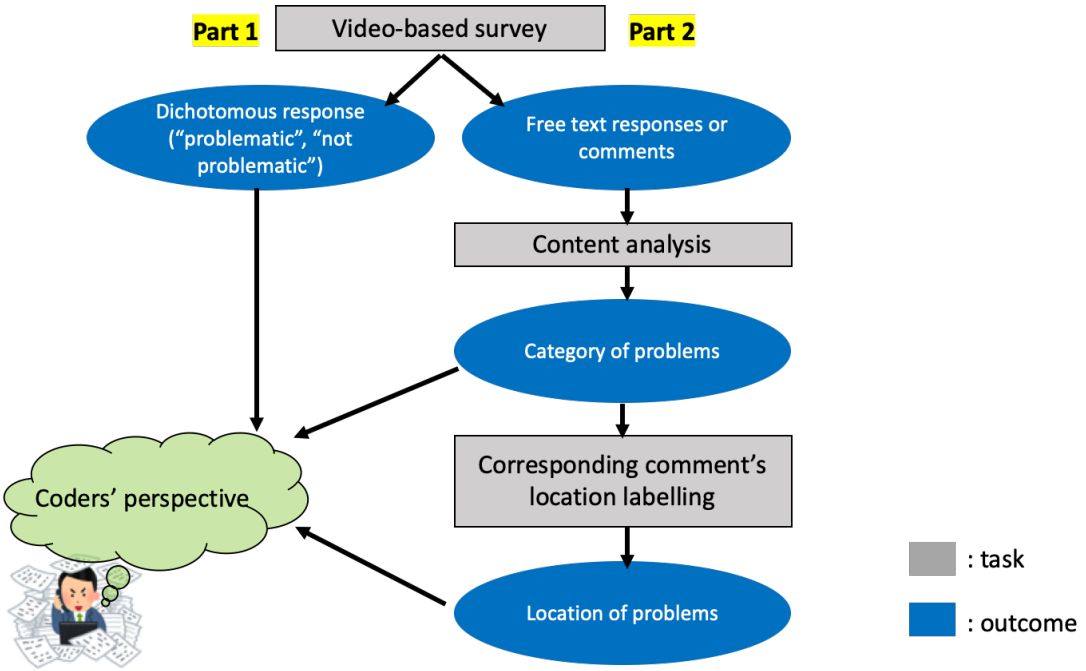

FIGURE 4.1: Design of video-based survey

4.2.1 Experiment material

Videos were embedded into PDF files to enable coders to record their comments and the time sequence of their comments. Eleven SOAP notes were used in this experiment. They were problematic pseudo SOAP notes in the Indonesian language (Appendix B) prepared by the experimenter, who is a medical coding lecturer. Figure 4.2 shows an example one of the pseudo SOAP notes used in the experiment. A SOAP note typing video was embedded into a single PDF file as a trial, with the other 10 videos divided evenly into two PDF files. The supporting materials included a readme file that contained instructions on how to perform the experiment in the rundown format and a video tutorial that explained how to assess and comment on the video using the commenting feature in Adobe Acrobat.

4.2.2 Experimental procedure

The experiment was conducted in a computer laboratory in Indonesia. Each coder used a desktop computer with a Windows operating system and connected to the internet. The survey was taken by five professional coders working at four different Indonesian hospitals. Two coders work in a primary level hospital, while the other three work in a secondary level hospital with a wider range of coding tasks. Table 4.1 shows the work experience of the five coders. The experimenter remotely supervised the experiment

```
dikatakan setelah minum gabapentin kedua tangan bengkak, nyeriseluruh badan,

TD 90/50
Thorax : SDV +/+N, Rh -/-, Wh -/-
Abd : BU+N,NT-

Neuralgia post herpetik (L)

Gabapentin dihentikan dl

PULANG :

NEURODEX NO X
S2DD TAB 1
-
ANALSIK NO X
S3DD TAB 1 PC
```

FIGURE 4.2: Example of SOAP note in Indonesian language

from Japan using Skype [52], while another experimenter supervised the experiment in situ.

TABLE 4.1: Working information of the coders

Participant	Working experience (years)	Coding cases per day	Hospital type
Coder 1	2	300	Secondary
Coder 2	9	320	Secondary
Coder 3	4	150	Primary
Coder 4	7	800	Secondary
Coder 5	5	150	Primary

First, coders were initially asked to identify the problematic SOAP notes after watching the SOAP note typing videos in the PDF files. Coders opened the PDF files using Adobe Acrobat software, played the video and labeled SOAP notes on the video as "problematic" or "non-problematic". The summative assessments made by the coders in were analyzed to check whether coders' general perspective correlated with each other and whether they could be broken down into a detailed perspective.

Next, coders were asked to provide comments about the SOAP notes. The comments could be about problems such as "the diagnosis is not specific", or not about problems such as "this information is helpful for coding". The comments were placed in the specific part of the SOAP notes by replaying the SOAP note typing videos, pausing the videos at specific parts of the videos and inserting comments. This enabled categorization of each coders' comments and their time sequence. Figure 4.3 gives an example of commented video (translated from Indonesian to English).

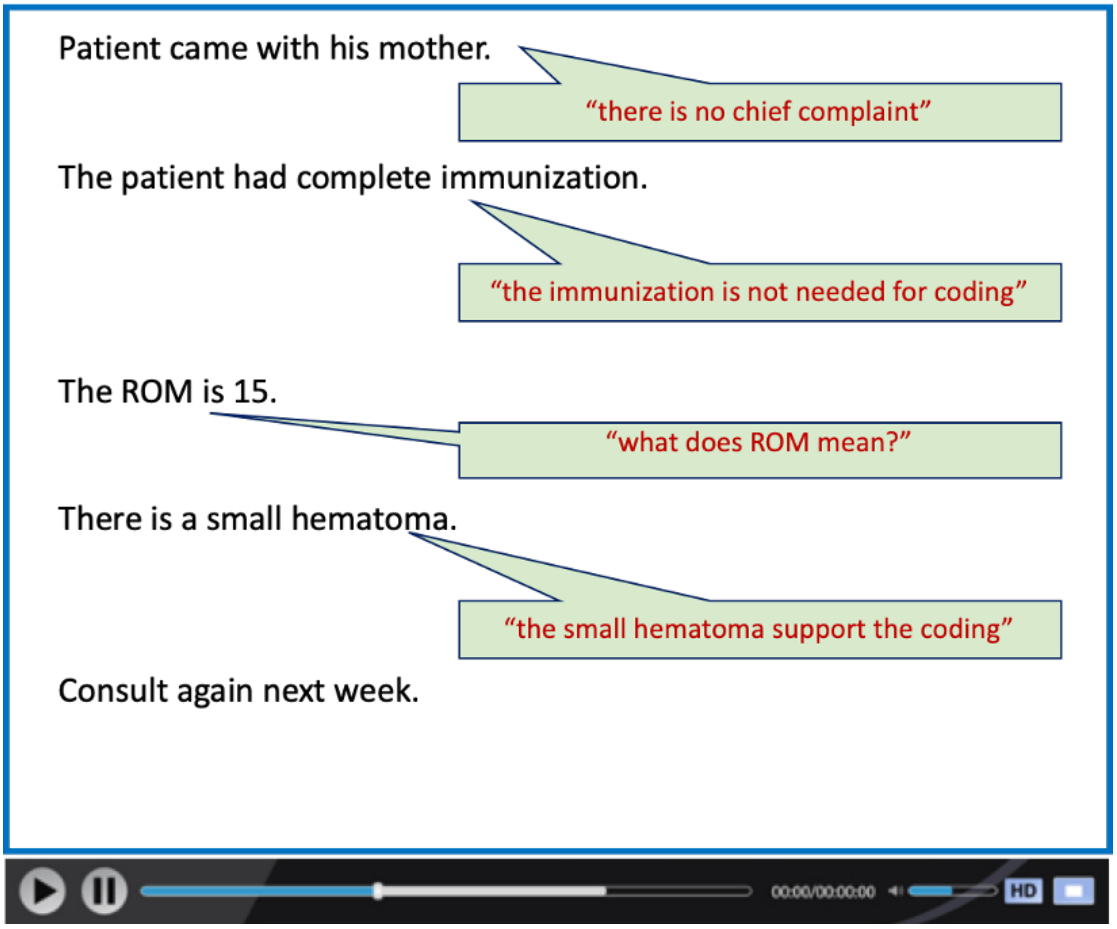

FIGURE 4.3: Example of commented video (translated)

4.2.3 Results

The five coders each assessed all 11 SOAP note typing videos. Of the 55 assessments, 44 (80%) were regarded by the coders as "problematic". Figure 4.4 shows the distribution of the SOAP notes regarded "problematic" and "non-problematic". Although all 11 SOAP notes were regarded as "problematic", the coders who work in the secondary level hospital evaluated some as "non- problematic".

The five coders provided a total of 166 comments. Based on keywords, these comments could be separated into four categories [53], abbreviations, incomplete, helpful and other. Comments such as "what does ROM stand for?" and "text is not understood" were classified into the "abbreviation" category problems. Comments asking about missing data, such as "where is the disease history?" and "there is no chief complaint" were classified into the "incomplete" category. Comments that were positive or encouraging, and which could help the coder perform accurate coding, such as "the small hematoma supports the coding", were classified into the "helpful" category. Comments such as "immunization is not needed for coding" were classified into the "other" category.

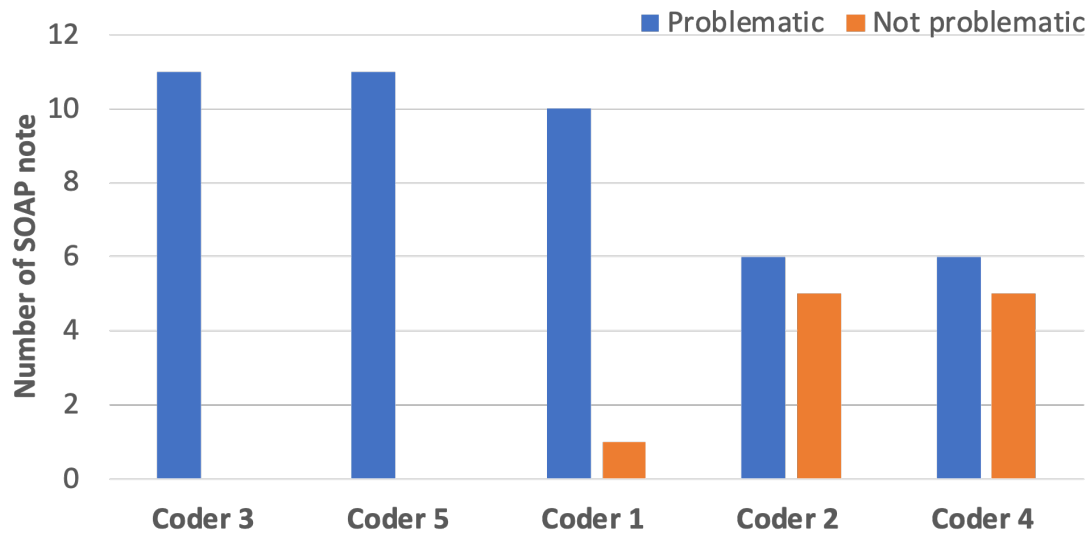

FIGURE 4.4: The "problematic" and "non-problematic" labels distribution

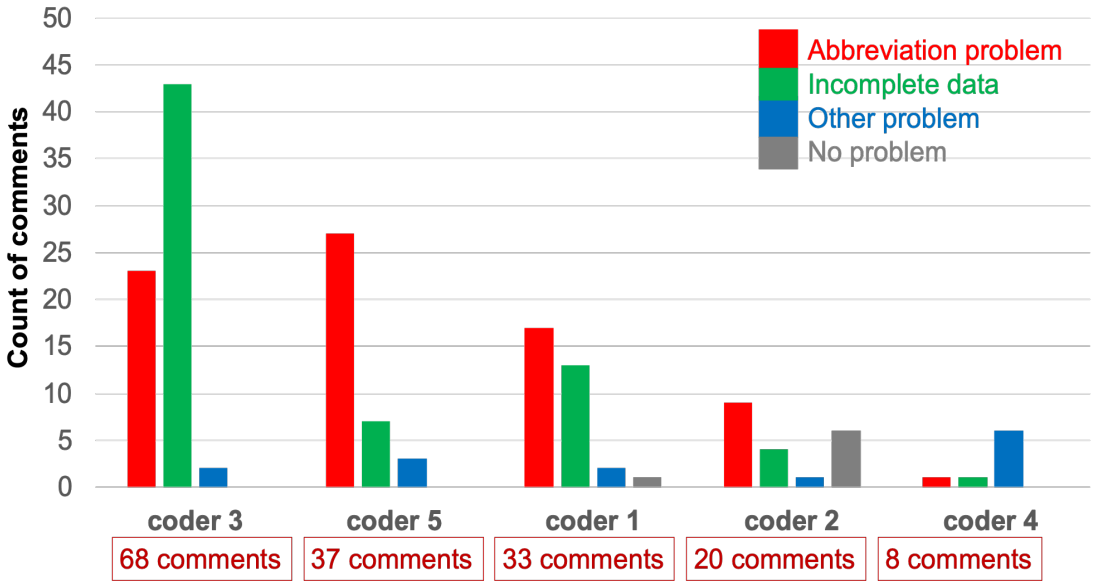

FIGURE 4.5: Comments categories distribution

Figure 4.5 shows the distribution of the comments by categories among the coders. Of the 166 comments, 77 (46.4%) were classified into the "abbreviation" category and 68 (41.0%) into the "incomplete" category.

Figure 4.6 shows the locations of comments by categories in the SOAP note typing videos. Before labeling the location of each specific comment location, each SOAP note was manually separated into its four parts, S, O, A, and P, and the comments assigned to each specific part by the coders. Of the 166 comments, 41 (24.7%) were assigned to S, 49 (29.5%) to O, 46 (27.7%) to A, and 30 (18.1%) to P. These comments included 113, 116, 65, and 185 words, respectively. The ratio of the number of comments to

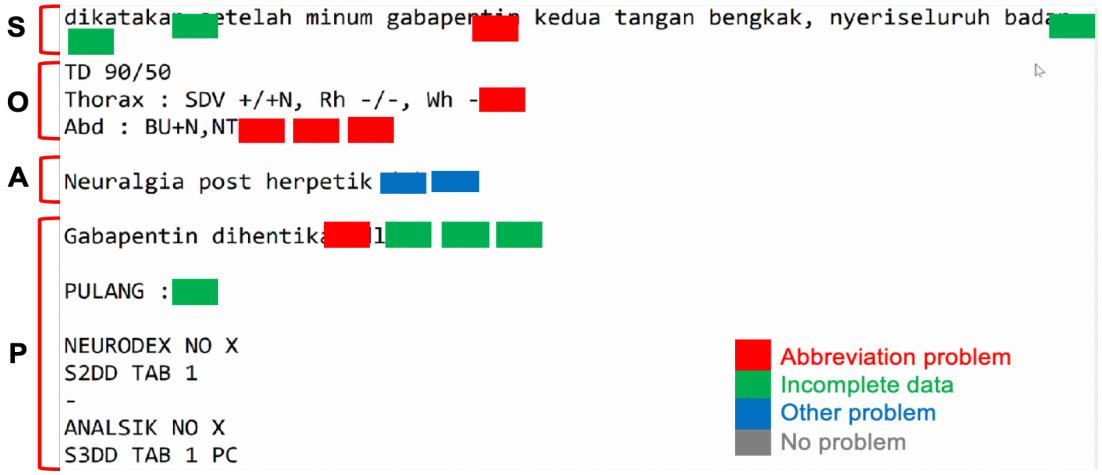

FIGURE 4.6: Examples of locations of the four categories of comments from the SOAP note typing videos

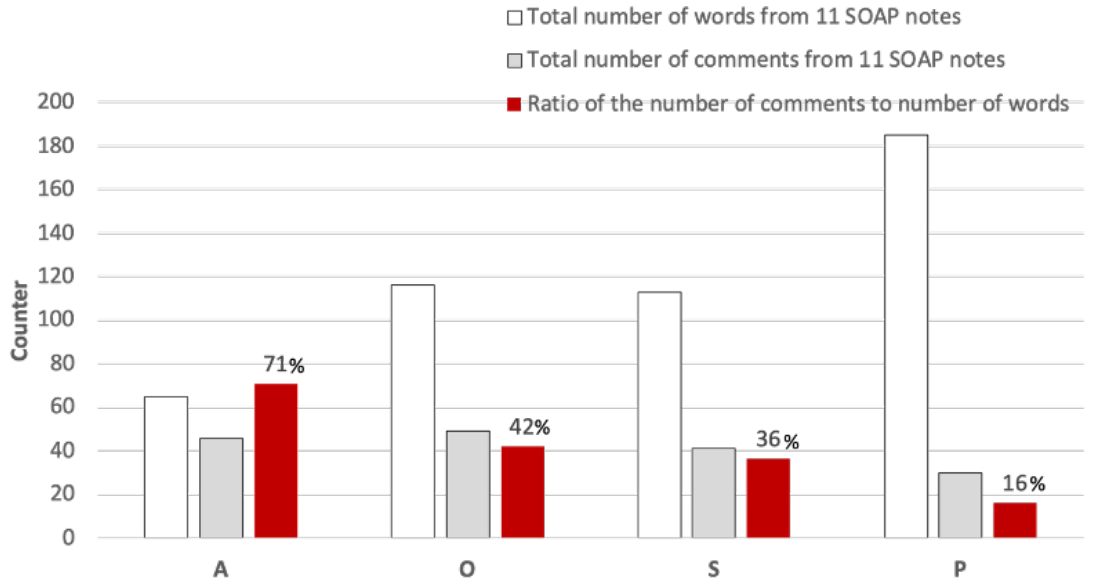

FIGURE 4.7: Ratio of the number of comments to the number of words made by coders

the number of words was calculated to determine the relative problematic content of the four parts, with a higher ratio indicating that the part was more problematic (Fig. 4.7). It was considered that the content of SOAP parts with more words is easier to interpret. It was also considered that a larger number of problem comments indicate that the part was more problematic. Part A had the fewest words, and was therefore less interpretable than S, O, and P. Part A had the second highest number of problem comments, below part O which had more words. Based on a comparison of the ratios of number of comments to number of words of each part, part A was the least interpretable and the most problematic part of the SOAP notes for the coders.

A map showing the percentages of the four problem categories in each part of the SOAP

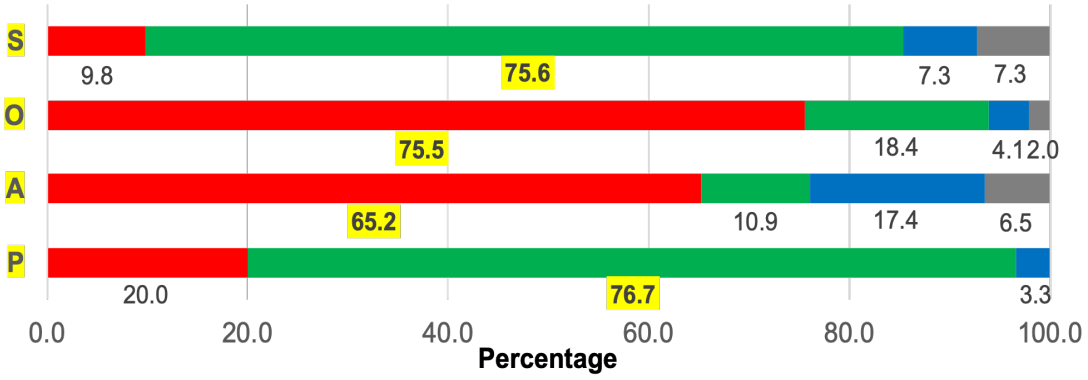

FIGURE 4.8: Percentages of the four problem categories in each part of the SOAP note

notes was also created. Figure 4.8 showed that there were specific problem categories in each part of the SOAP notes. For example, 31 (75.6%) of 41 comments regarding the S part of the SOAP notes and 23 (76.7%) of 30 in the P part were about "incomplete" notes, whereas 37 (75.5%) of 49 comments in the O part and 30 (65.2%) of 46 in the A part were about abbreviations or text that was not understood.

4.3 Proposal of agent's features

The present study confirmed SOAP notes are frequently problematic, with incomplete data [54] and abbreviation problems [55] often encountered by coders. These problems correlate with the informal writing style of most physicians, making it difficult for readers to understand the notes [56].

Figure 4.9 shows agent's features based on the coders' perspective. A multi-feature agent-based system has been proposed to support or guide physicians in writing SOAP notes, based on the structures required by coders to produce accurate medical codes. A detailed description of each feature follows.

4.3.1 Recognition of "abbreviation" part

Based on Fig. 4.8, the coders agreed that most of the problems in the O and A parts of the SOAP notes were caused by abbreviations or text that was not understood. The coders' perspective was used by the agent to focus on recognizing these abbreviations and text in the O and A parts of SOAP notes. Physicians frequently utilize abbreviations in the O part to describe the results of their physical examinations. Although abbreviations are standardized in most hospitals [57], physicians may be unaware of or forget some of

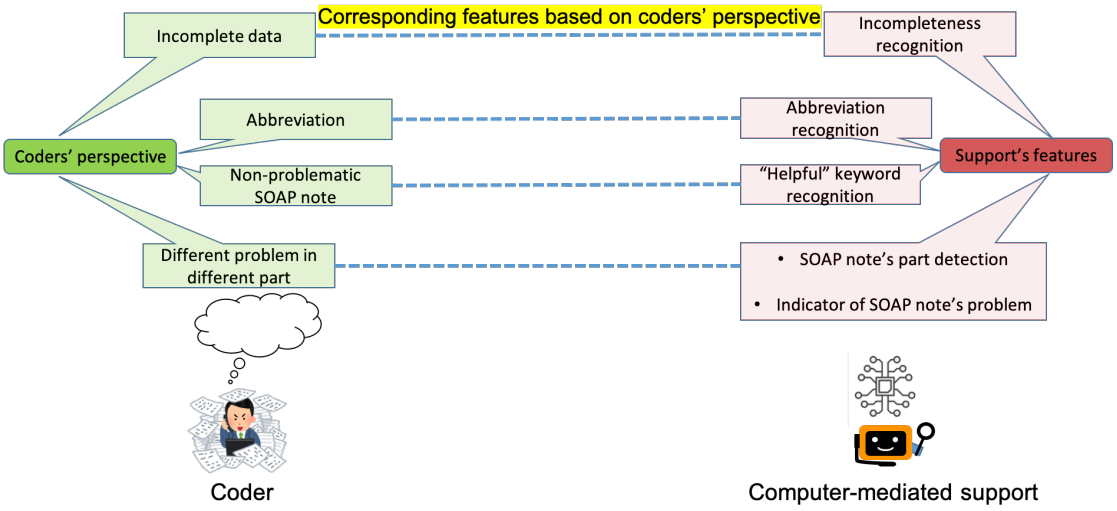

FIGURE 4.9: Agent's features based on the coders' perspective

these standard abbreviations while working. The agent should be designed to recognize non-standard abbreviations and change them to standard abbreviations.

Upon presentation to the physician, the physicians can accept or decline the standard abbreviation. In part A, physicians often use diagnostic terms or text that may not be understood by coders. Coders expect that the text dealing with diagnosis will be similar to standard diagnostic terms in the ICD [58]. Coders therefore have difficulty putting language and expressions used by physicians into terms matching the ICD diagnostic codes [59]. The agent should be designed to recognize the physicians' diagnosis text and match it with similar terms in the ICD standard and present it to physicians. The physicians can accept or decline this standardized diagnosis.

The agent prompts physicians to use standard medical abbreviations/words only when rare/advanced medical abbreviations are not in the list of abbreviations that are accepted for use in SOAP notes. Otherwise, the agent directly auto corrects the abbreviations that are in the list of accepted abbreviations.

4.3.2 Recognition of "incomplete" part and auto-completion support

Based on Fig. 4.8, the coders agreed that most of the problems in the S and P parts of SOAP notes were caused by incomplete or non-specific data. This coders' perspective was used by the agent to complete the missing information in part S and to complete the medical procedure text in part P, making them more specific in accordance with ICD standards. The S part of the SOAP notes contains sub-parts, which reflect the patient's subjective complaints and history. This subjective information is used by physicians to choose relevant physical or other examinations in O. If some sub-parts in S are missing,

the examination may not be sufficiently specific, resulting in a non-specific diagnosis. Completion of the sub-parts in S can improve the accuracy of the medical codes [60]. The agent supports the physicians in completing these sub-parts by providing a sub-part completion system, which is similar to a template completion system [61]. It is activated by the agent if the agent detects a missing sub-part. The P part contains a description of medical procedures. However, physicians may indicate a non-specific medical procedure [62]. The specificity of ICD codes is based on the specificity of medical procedures indicated by the physicians, resulting in greater reimbursement of hospitals for insurance claims and enhancing hospital cash flow management [63]. The agent helps the physician record specific medical procedures by providing a text completion system. This system presents the physician with the most likely standardized medical procedures based on the ICD standard.

The auto-completion support is commonly deployed in word processor and virtual keyboard applications. The problem with common auto-completion is that the predicted text is not always 100% accurate. In some cases, users do not want to use the auto-completion support. One idea is to design an auto-completion support that recognizes when the users or physicians need to use it. With the intended purpose, the predicted text by auto-completion need not be necessary to be 100% accurate because it is highly likely that physicians will correct wrong predictions. One aspect to consider is the willingness of the physician to use auto-completion support when typing or during pauses when typing the medical notes. The basic design for such a kind of auto-completion system is an agent-based completion design.

An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives [64]. An agent is capable of independent action and its action has a purpose. Agents have been used in various application areas, and particularly in two areas:

1. Distributed systems (processing nodes)
2. Personal software assistants (aiding a user)

Figure 4.10 shows the typical agent-based design.

Agent-based completion is a word/sentence completion system in which an agent monitors physicians' input behavior and supports them in filling the necessary parameters for a data collection task. An example of a system that monitors user behavior is Amazon.com's book recommender system [66].

An agent-based completion system is one that represents the coders in the task of collecting parameter-text pairs from the medical notes. The agent interacts with the physician

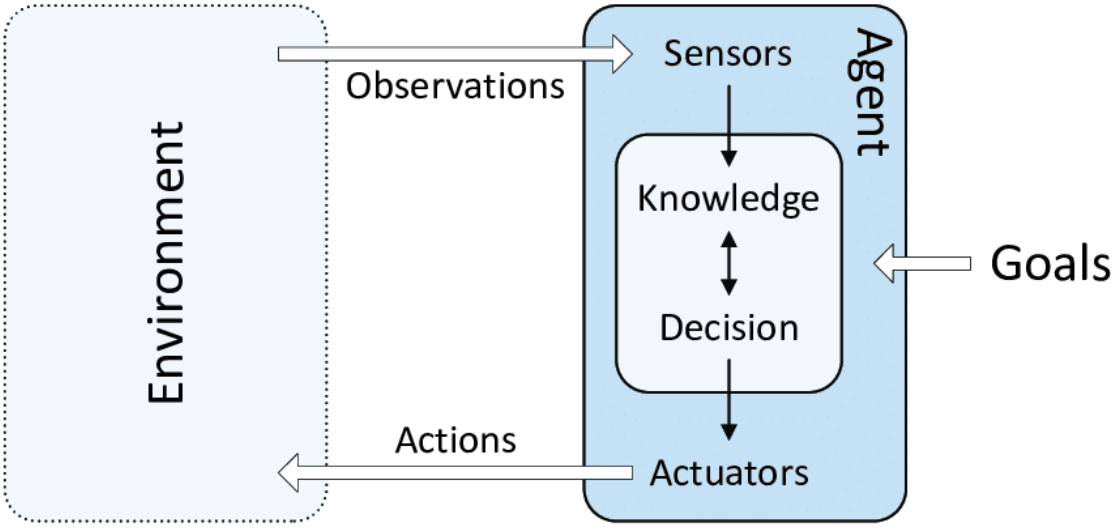

FIGURE 4.10: Agent-based design
[65]

while typing the medical notes by recommending the next parameter to fill. The main features of the agent are: 1) It understands the parameters to be filled from the perspective of the coders, and 2) It can recommend the next parameter to be filled based on the physician's way of typing the narrative text. By combining these two features, the agent would be capable of completing the medical notes by interacting with the physician. The agent-based completion supports the task of collecting parameter-text pairs by monitoring the physician's inputted text and time gap between the inputs. Figure 4.11 shows our agent-based completion framework. The agent interacts with physician's inputted text and physician's typing pause time or time gap. These two input signals trigger the agent into cooperating with the physician in the task of collecting parameter-text pairs from the medical notes.

For the first signal, which concerns the narrative text, the agent conducts a categorization procedure, using the Named Entity Recognition (NER) algorithm [67]. The NER algorithm categorizes the unknown text into the particular medical note's parameter by matching [68] the similarity of unknown text with the known text of a particular parameter. If the time gap between the physician's inputs exceeds a certain threshold, the agent recommends the next parameter to be filled.

As for the recommendation, a knowledge base is set up for the agent. The knowledge base is built up by extracting the parameters as graph vertices, and the order of the parameters as graph edges from the collection of medical notes training dataset that can be accessed from https://www.mtsamples.com/. Figure 4.12 shows a pre-trained directed graph consisting of interconnected parameters. Each parameter has its own

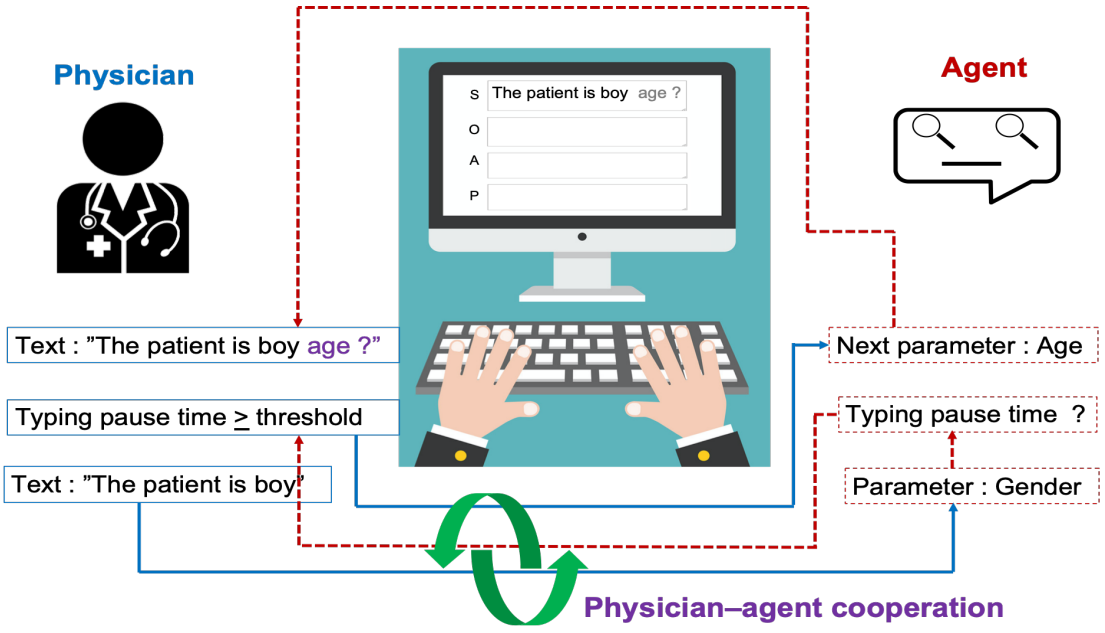

FIGURE 4.11: Agent and physician cooperation bridged by a narrative text interface

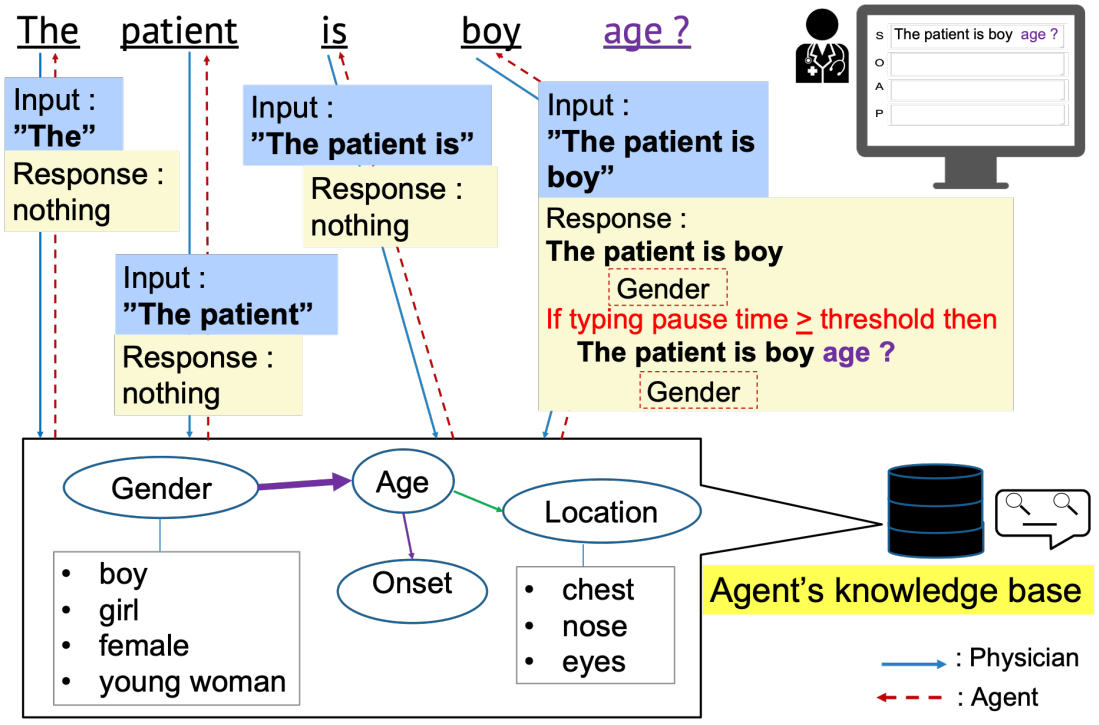

FIGURE 4.12: Interaction scenario for collecting the medical note parameters

related keywords, for example, the parameter gender consists of “boy”, “girl”, “female”, “young woman”.

Furthermore, Figure 4.12 shows an example of an interaction scenario between a physician and the agent. In this scenario, the agent attempts to categorize the inputted text into one of the medical note parameters. If the agent did not succeed, then it will not make a response. If the agent succeeds, the inputted text will be tagged with the parameter identifier. For example, “the patient is boy” is tagged with the “gender” parameter. While categorizing the inputted text, the agent also monitors the time gap between the physician’s inputs. If the gap is greater than a predefined threshold after categorization, the agent recommends the next parameter to be filled. For example, after the “gender” parameter, the agent will recommend the “age” parameter. This recommendation is based on the pre-trained directed graph, as shown in Figure 4.12, in which there is a direction horizontally from the “gender” parameter to the “age” parameter.

4.3.3 Identification of parts of SOAP notes

In our case, the SOAP notes were not explicitly separated into four parts. Ideally, SOAP notes should be divided into four text boxes [69]. If these four parts are explicitly separated, the agent simply activates the related features without identifying the position on the notes containing that designation by the physician. If the problematic text is in the S text box, the agent activates the sub-part completion system. If the problematic text is in the O text box, the agent activates the abbreviation recognition system. However, in the absence of explicit separation of the text boxes as shown in Fig. 4.6, the agent will be unable to determine which feature to activate. Therefore, the agent has a feature to identify the different parts of the SOAP notes. Using this feature, the agent can support the physician in writing SOAP notes that conform with the structures required by the coders.

4.3.4 Indicator of SOAP note problems

This feature is based on findings by coders when SOAP notes are problematic, with comments by the coders supporting these findings. Also, one coder assessed problematic SOAP notes as non-problematic without providing any comments as shown in Figs. 4.4 and 4.5. This indicator is useful when physicians who type SOAP notes with support from the agent want to submit their SOAP notes and want to know whether they remain problematic. This indicator constitutes a type of progressive feedback report to physicians, which increases their motivation to write non-problematic SOAP notes [70].

4.3.5 "Helpful" keyword identification for encouraging physicians

Some coders assess SOAP notes as being generally non-problematic, even when containing comments to the contrary, as shown in Fig. 4.4 and Fig. 4.5. These findings suggest that the agent should identify "helpful" keywords that are positive or "non-problematic" comments in the SOAP notes. If the system is able to classify problematic SOAP notes as being non-problematic by detecting many helpful keywords in the commentary of coders at a secondary level hospital, the system could notify physicians that some of their texts are helpful for the coders. It means that some notifications will be encouraging rather than assertive. "Helpful" keywords identified by coders can be used to classify whether a SOAP note is problematic or not. The feature has a novel "encouraging" characteristic providing positive feedback to physicians, while also helping coders produce appropriate codes.

4.4 Summary

This study had several limitations. First, this study included only five coders. Second, the SOAP note typing videos used in this study were made from pseudo SOAP notes, which may not represent actual SOAP notes in Indonesian hospitals. Finally, the coders' comments were not well-distributed, with one coder providing 41% and another providing only 5% of the total comments. The remaining comments (54%) were provided by the three other coders. In this study, the detailed comments about SOAP notes with the small number of coders were collected. Based on the results with this study, quantitative analysis with a larger number of subjects and concerning general claims in the real SOAP notes in Indonesia could be conducted in future work.

This study can be applied to other countries with the same situation as Indonesia, including the use of coders for medical coding tasks and a DRG system for insurance claims. For example, countries such as Thailand and Malaysia have the same problem with medical coding [71] as Indonesia.

This study assessed problems with SOAP notes from the perspective of medical coders. The coders showed 80% agreement in their assessments of problematic SOAP notes. Evaluation of their comments showed that the two most frequent problems (87%) faced by coders when encountering physicians' SOAP notes were abbreviations or text that they could not understand and incomplete or non-specific data. Coders also agreed that each part of SOAP notes has its own problems. Solving these problems requires an appropriate agent with the features proposed in this study. Evaluation from the coders' perspective can help to develop an agent that can address specific parts of SOAP notes.

For example, in the O part it can focus on developing an abbreviation recognition system, and in the S part it can focus on developing a sub-part completion system. The agent should also be able to identify parts of SOAP notes when they are not explicitly separated by text boxes. This study is able to understand the coders' perspective for developing features of the agent-based writing support system.

Video-based surveys provide many benefits over "traditional" in-person surveys, including [49]:

1. Speed – Running an in-person study can take weeks or months, while video surveys can capture results in a matter of days.
2. Reduced bias – In-person surveys suffer from a number of biases, such as observer bias, interviewer bias and social desirability bias. Video surveys significantly reduce these biases.
3. Authenticity – Since respondents are recording their replies from the comfort of their homes, offices or on the go, the interviewer tends to get more "authentic" responses compared with facility-based in-person surveys.
4. Scale – Video surveys enable the interviewer to conduct many more interviews in a shorter period of time compared to using real-time in-person surveys. The interviewer can get an acceptable sample size in a brief period of time using video surveys.
5. Easier and faster analysis – The analysis of the feedback from the video survey is easier and faster due to the structured questionnaire format and the way each reply is attached to a specific question.
6. Reduced cost – The costs of running an in-person study can be substantial. The costs for renting facilities, paying high incentives to recruit respondents to a central location, providing refreshments, travel, etc. can add up quickly. Video-based surveys can be cheaper, while producing the same or even higher quality outcome.

Given the advantages of the video-based survey, it is suitable for understanding perspectives of a range of healthcare workers including coders, physicians, patients, nurses, midwives, pharmacists and laboratory technicians. It is suitable because in the healthcare domain, privacy is a major concern when collecting information from healthcare entities [72]. Using video-based survey, the face or some part of the participant's body can be blurred, which may be more privacy-concerned. With the privacy concern, data can be more difficult to acquire and takes more effort with in-person surveys. Therefore video-based surveys are appropriate to bridge the gap. With video-based surveys, the

interviewers or experimenters collect more data with a smaller budget and get a more reliable and higher quality outcome, while still maintaining the healthcare entity's confidentiality. A healthcare entity in this case is an entity who is involved in the healthcare services, such as physician, nurse, patient, and coder. If more relevant data are collected, more flow among healthcare entities in hospitals could integrate to produce a comprehensive support system, which not only focuses on the physicians-coders interaction but also on other healthcare entities.

The coders revealed that the physician's way of writing is problematic for them. From the coders' perspective, the physicians should write medical notes using standardized and structured clinical terms easier for coders to decipher and transform unstructured texts into coded data. However, physicians use informal expressions, such as those found in tweets or other informal texts that are often used in Social networking services (SNSs) communication, rather than standard language and format including proper punctuation, spelling, spacing and formatting or standardized abbreviations [73]. One type of informal text is informal abbreviation, commonly used by physicians to express patients' diagnoses. While informal abbreviations help physicians write efficiently from their perspective, coders struggle to understand them. From the perspective of data interoperability, like semantic data interoperability, the variety of informal abbreviations used by physicians may impact the advantages of EMR as a digital tool for primary and secondary usage of clinical data processing.

Therefore, a way of detecting and handling of informal abbreviations was developed to aid coders. It is evaluated in Chapter 5.

Chapter 5

Recognition Design of a Support System for Writing Medical Notes

This chapter describes deep learning modelling using LSTM to detect *informal* abbreviations in medical notes. This is a vital task for structured entity identification in EMR. This chapter was published in EJBI (Appendix F).

5.1 Overview

EMR that use structured data elements document patient information using controlled vocabulary rather than narrative text. The historic lack of structured data and standardization in the healthcare industry still causes problems when EMR content is shared among providers. Currently, the healthcare industry is far from having the desired set-up whereby patients have one complete, accurate EMR from which the quality of their individual health care can be monitored and maintained.

In many cases, critical data elements, such as problem lists, medications and allergies, are inconsistent across EMR systems. The coding of these data elements often differs from one health system to another, even when using the same EMR. In other instances, uncontrolled vocabulary usage is drawn from a variety of sources of clinical definitions such as SNOMED (Systematic Nomenclature of Medicine), a hierarchical medical classification system, or ICD-9, ICD-10, and ICD-11 [74].

Many health care initiative programs require structured data. All require a robust level of reporting to measure the quality of clinical data and provide clinical decision support tools to physicians and hospitals. Accurate documentation helps provide a baseline of the cost of patient care, and it can affect profitability. It is essential that physicians

accurately document all of their work so that they not only provide a complete medical record of delivery of care but also receive full and fair reimbursement for their services. Accurate coding can also offer some protection during audits. On the other hand, inaccurate coding can have unintended and deleterious consequences [75].

Coding involves charting and documenting the service provided during a patient's visit. Coding is required to combine rich clinical data from electronic medical record systems running in provider sites across the county into large patient cohorts. There are data quality and semantic interoperability requirements that must be met prior to the combining of the clinical data [76], such as the requirement to have common clinical vocabulary, which could be achieved by having structured and standardized data.

Without these standards, consolidation and harmonization of rich clinical data will yield only a portion of the expected value. Identifying genetic determinants of disease from this large population, for instance, requires a clinical data set based on standards such that large sub-populations can be compared using common clinical vocabulary [77]. The standards to be addressed fall into two categories: 1) Existing, popularly used or incentivized coding systems and 2) value lists that need to be defined.

Category 1 includes coding systems such as: ICD-10 diagnosis codes (and historically ICD-9), CPT and ICD-10 procedure codes, LOINC codes for laboratory orders and results and RxNorm codes for drugs. The ICD and CPT coding systems have been used in EMRs and billing systems for decades.

Category 2 data include critical items used for cohort subdivision such as: vital signs (height, weight, blood pressure, etc.); allergies; smoking history; demographics such as race, ethnicity, address, age, socio-economics; and encounter types, notably inpatient, outpatient, home health, and so on.

The second category is the most challenging, given the lack of common national standards in these areas [77]. This means that the organizations sponsoring the creation of the cohort must define standards in advance and provide the funding to the contributing providers to map their local data to those standards. In addition, a common method of avoiding duplicate records for the same patient should be devised by each sponsor to ensure data from the same patient across multiple contributing sites is linked to the same person in the cohort. Lastly, unstructured text extraction software and models should be made by the sponsoring organizations so that all examination notes, pathology reports, radiology reports, etc. can be processed to extract or derive clinical facts from the rich collections of unstructured text in every contributing EMR.

While the current efforts are daunting, they are critical to the success of any attempt to combine clinical data from multiple providers into a common repository. Discoveries

from large populations of patients' genetic variant data cannot be made without common clinical vocabulary and ontologies that enable consistent profiling of millions of patients.

Medical data must be structured for semantic interoperability and to facilitate data processing for primary and secondary usage. Semantic interoperability is essential for the primary and secondary usage of clinical data in EMR since a seamless communication platform is desired, rendering data compatible whenever a patient migrates from one physician to another [78, 79]. Semantic interoperability ensures that the meaning of medical concepts can be shared across systems, thus providing a digital and common language for medical terms that is understandable to humans and machines.

For instance, the sentence "patient g2-p2 experiences asthma attack" includes information about pregnancy history, in the form of the abbreviated term "g2-p2". If the same sentence were written using a standard for semantic interoperability, such as SNOMED CT [80], the abbreviation "g2-p2" would be replaced by "gravida 2 or second pregnancy and para 2 or parity 2". In the abbreviated sentence, if the term "g2-p2" was not detected by a healthcare information system that employs SNOMED CT, such as a Clinical Physician Order Entry (CPOE), the person responsible for processing the medication order might misunderstand or fail to recognize it, leading to an erroneous interpretation and inappropriate drug administration, which increases the risk for the patient.

Additionally, a large amount of medical data currently produced is for free text medical notes [81], such as SOAP notes and discharge summaries. These documents are often created in varied writing styles [82], using informal abbreviations that do not follow standard medical nomenclature [83]. Since these documents are implicitly composed in an unstructured format, it is necessary to parse them and identify specific information in the text in order to generate structured data that can be shared across systems. In this process, detecting abbreviations correctly is a key requirement since terms unknown to the system could be ignored or misrepresented in the final output.

5.2 Agent's feature: abbreviation detection

Abbreviation detection, shown in Fig. 4.5, is an important feature for an agent from the coders' perspective. Healthcare professionals use abbreviations as a convenient way to represent long biomedical words and phrases. Those abbreviations often contain important healthcare information (e.g., names of diseases, drugs and procedures), which must be recognizable and accurate in health records. Nevertheless, studies [84] have shown that confusion caused by frequently changing and highly ambiguous abbreviations

impedes effective communication among healthcare providers and patients, potentially diminishing healthcare quality and safety.

Detection of abbreviations falls under a natural language processing task, named entity recognition (NER). NER identifies specific terms for medical concepts within the text [85]. In our case, it identifies the presence of abbreviations. This is done in two steps: abbreviation detection and then abbreviation disambiguation, which is a special case of word sense disambiguation (WSD) [86]. In this study the focus is on abbreviation detection, which remains a challenging problem due to the variability of abbreviations.

To detect informal abbreviations, an LSTM based model [87], which is a form of deep learning technique, is proposed. Deep learning algorithms – a subset of machine learning methods – currently offer the best performance in tasks that involve learning from sequential data, such as free text medical notes. There are several deep learning algorithms for the information extraction task, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Recursive Neural Network, Neural Language Model and Deep transformer [88]. The information that can be extracted from EMR using deep learning is varied, from a patient phenotyping using CNN [89] to named entities using RNN [90].

In our informal abbreviation extraction study, an RNN variant, an LSTM based model, was chosen as our proposed model. It learns without the need of prior feature extraction processes [91]. Although the feature extraction task is particularly difficult in the case of abbreviation detection, the LSTM model can be effective, since it also takes into account the context in which sequences occur when detecting the intended entity at each location. In other words, the model can detect abbreviations by exploring their surroundings. For example, the LSTM model is able to detect that "G3P3" is an abbreviation in the sentence *"pregnancy-labor-spontaneous delivery to a G3P3"*, as is the case of "g3.p2A1" in the sentence *"pregnancy-labor-spontaneous delivery to a g3.p2A1"*. It does so by understanding that these words occur in similar contexts.

This study presents a deep learning model using LSTM to detect *informal* abbreviations in medical notes, which is a vital task for semantic interoperability in EMR. Additionally, our approach expands the field of application of the LSTM model since the technique has not been used before for this kind of task. Figure 5.1 shows our motivation in detecting informal abbreviations to optimize the medical coding task.

A previous study [92] has shown the potential utility of machine-learning-based (ML-based) abbreviation detection methods for predefined abbreviations in English texts and medical notes. That study focused on detecting formal abbreviation with predefined extracted features using traditional machine learning algorithms, such as decision trees,

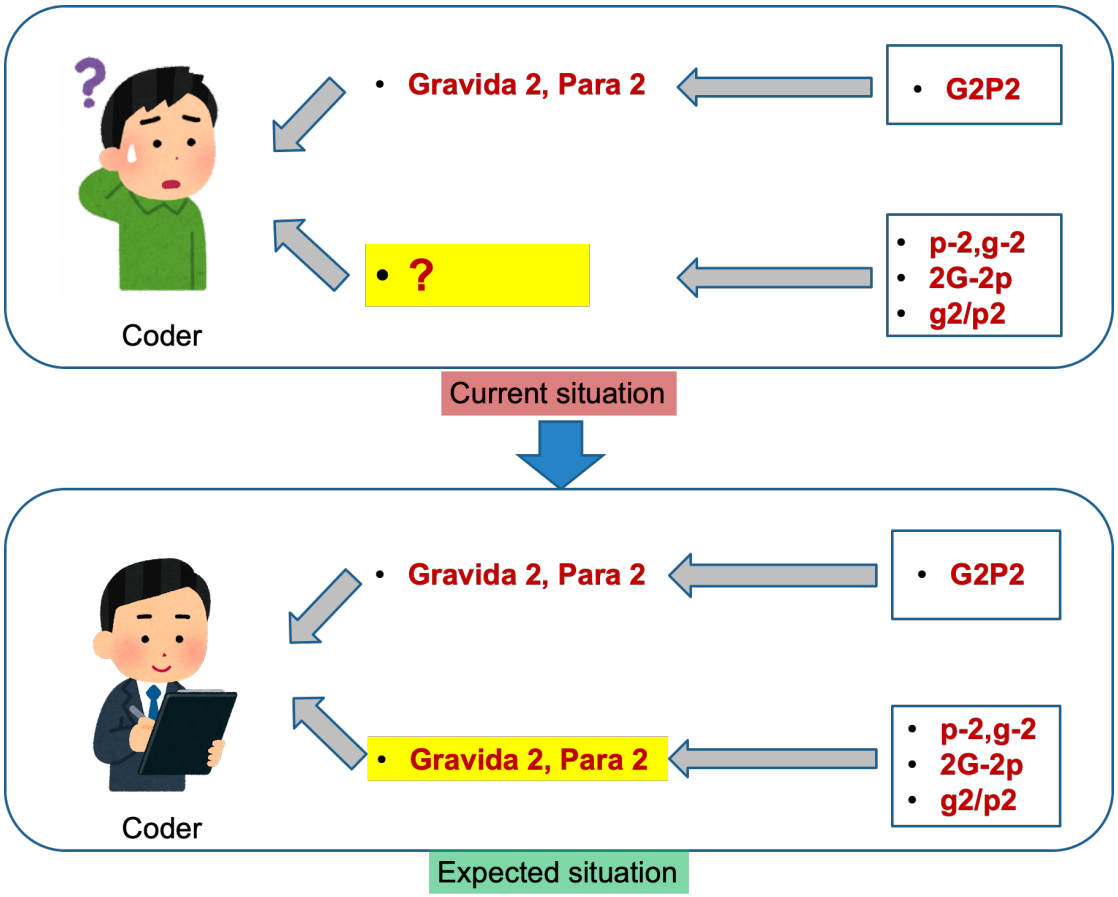

FIGURE 5.1: Motivation in detecting informal abbreviations

random forests, and support vector machines. Another previous study to detect abbreviations also used a machine learning algorithm, such as stochastic gradient descent [93]. Meanwhile a deep learning algorithm such as bidirectional LSTM and CNN are used to detect more complex named entities such as a nested named entity [94] and a biomedical named entity [95]. In contrast, our study focuses on detecting *informal* abbreviations as a noisy entity. This has not been reported in previous studies.

5.3 LSTM-based informal abbreviation detection

In order to detect informal abbreviations in free text medical notes, an LSTM based method was proposed. The proposed method has components corresponding to the following problems: First, detection of *noisy entities*, i.e., terms to be identified in a set of unstructured data. Second, the detection task suffering from class imbalance in which the occurrence of different types of entities in the data varies widely, causing some to be underrepresented. Free text medical notes are an example of this kind of data. A similar case is that of Twitter messages (tweets [96]). Both types of data consist of noisy

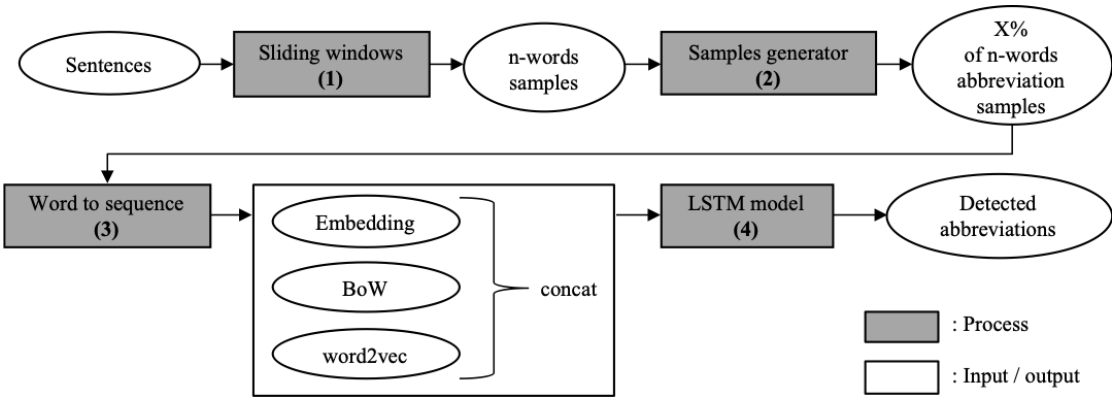

FIGURE 5.2: Framework for informal abbreviations detection using LSTM

entities that lack implicit linguistic formalism (*e.g.* due to having improper punctuation, spelling, spacing, formatting) while also including non-standard abbreviations [73]. The LSTM model outperforms other strategies when used for tasks of entity recognition in this kind of noisy data set.

Noisy entities are particularly difficult to detect if the surrounding context is not taken into account, including the order and relative distribution of the entity's occurrence. The entity's order can be represented by a word2vec matrix, while the entity's relative distribution can be represented by a Bag of Words (BoW) matrix. For this reason, the addition of these representations to the input of the model can increase its accuracy.

Additionally, the problem of class imbalance must be addressed. It happens when the total number of items in a data class (in this case, abbreviations) is far less than the total number of items in another class (non-abbreviation). To deal with this problem, one common approach is to use oversampling [97], which consists of increasing the number of items of the under-represented class, so it has a greater impact on the machine learning algorithm. Our method consists of two parts: *pre-processing*, with data oversampling to address class imbalance; and *processing*, with additional inputs to represent word context (word2vec, BoW). Figure 5.2 shows a detection flow diagram of our framework.

The pre-processing part is further divided into two steps, sliding window and samples generator. The first step (sliding window) involves breaking the longer and varied-length sentences into fixed groups of words. This is necessary because the LSTM model requires short fixed-length sequential inputs. This transformation also alleviates the problem of limited data sets (Section 5.7), since it increases the number of samples. The second phase of pre-processing (samples generator) increases the number of samples of infrequent data [98], to improve the LSTM model's prediction performance, since it performs better with more balanced data.

The strength of the LSTM model is that it learns its word embedding input automatically, although it can also learn from pre-trained word embedding. Pre-trained word embedding is represented as a matrix, which can be produced using a word representation algorithm, such as BoW and word2vec. In our abbreviation detection, word embedding with the pre-trained embedding, BoW matrix and word2vec matrix (step 3) was concatenated to achieve higher performance. This takes advantage of the fact that the LSTM model (step 4) can process multiple features for its learning [99].

5.3.1 Pre-processing

The pre-processing process was designed for solving the problems of class imbalance and unfair learning of the LSTM classifier. The issue causes a low predictive performance of classifier to recognize the minority class.

To tackle the class imbalance issue, a sliding window [100] to split the input sentence into chunks, and a sample generator to increase the number of abbreviation samples was used. The sliding windows step creates fixed-length samples with size n, consisting of a sequence of words from the sentence, without any changes in the words themselves. In our case, window size 5 (n=5), and step 1 (the next window starts 1 word to the right) was used. The input sentence *"coronary artery disease history of seizure disorder GERD bipolar"* is chunked into five fixed size samples, starting with *"coronary artery disease history of"*, then *"artery disease history of seizure"*, and so on. Words as abbreviation/non-abbreviation for the training process was manually labelled.

After fixed size inputs are produced, a sample generator is used to increase the ratio of abbreviation samples in the dataset. This step is necessary to create fairer learning by the LSTM model, by reducing class imbalance from the input. By specifying the percentage of samples that should contain at least one abbreviation, the ratio of abbreviation/non-abbreviation samples in the input presented to the model was determined.

5.3.2 Processing

The processing stage was designed with consideration of sentence feature extraction. In this stage, an LSTM with one embedding layer was used as our baseline model. The common approach when using an LSTM model for entity classification in text is to convert each word in the sample to a word index, which is a positive integer. After that, each word index is turned into embedding vectors of fixed size [101] before being presented for input into the LSTM model.

A bidirectional LSTMs algorithm is used for our detection of informal abbreviations. The bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance with respect to sequence classification problems [102]. In problems where all time-steps of the input sequence are available, such as free text sentences, the bidirectional LSTM trains two instead of one LSTM with regard to the input sequence. The first has training on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the model and result in learning the problem faster and even more complete.

The model was trained using additional pre-trained matrices, such as word frequency or BoW matrix [103] and word2vec matrix [104], to increase its performance. The BoW matrix represents the occurrence frequency of each word in relation to the complete vocabulary in the dataset. The word2vec matrix represents words as vectors in such a way that words which share similar contexts are closer in the vector space [105]. For example, the BoW matrix of the word "5" in the sentence *"A 5 yo boy brought by his parents because of 2 days of cough"* is (0,1,0,0,0,0,0,0,0,0,0,0,0,0) and the vector for the word "5" are clustered together with the vector for the word "2".

As the LSTM model can process multiple features to increase its performance [99], the additional matrices are concatenated with the embedding vectors of the baseline model. The BoW and word2vec matrices were used as additional input to the LSTM model based on the idea that abbreviations have frequencies and vectors in the word2vec vector space that differentiate them from non-abbreviations and ordinary English words. It can be expected that the model's performance would be increased with the inclusion of more relevant pre-trained features, represented by these additional matrices.

In summary, our basic LSTM model used only embedding vectors as input. Evaluation of the other input combinations expanded this basic model by including additional pre-trained features in the form of the BoW and word2vec matrices. To enable this additional input into the LSTM model, it was necessary to include additional layers for each of the additional matrix of features [105]. Finally, all the LSTM layers in the model are concatenated to predict the abbreviations in the sample using LSTM hyperparameters, such as dropout, dense and activation layer. In our model, the hyperparameters are the same for all input combinations. Figure 5.3 shows our LSTM model.

5.4 Experiment

Experiments were conducted to verify the performance of the proposed LSTM-based method. The experimental conditions were designed to confirm the effectiveness of

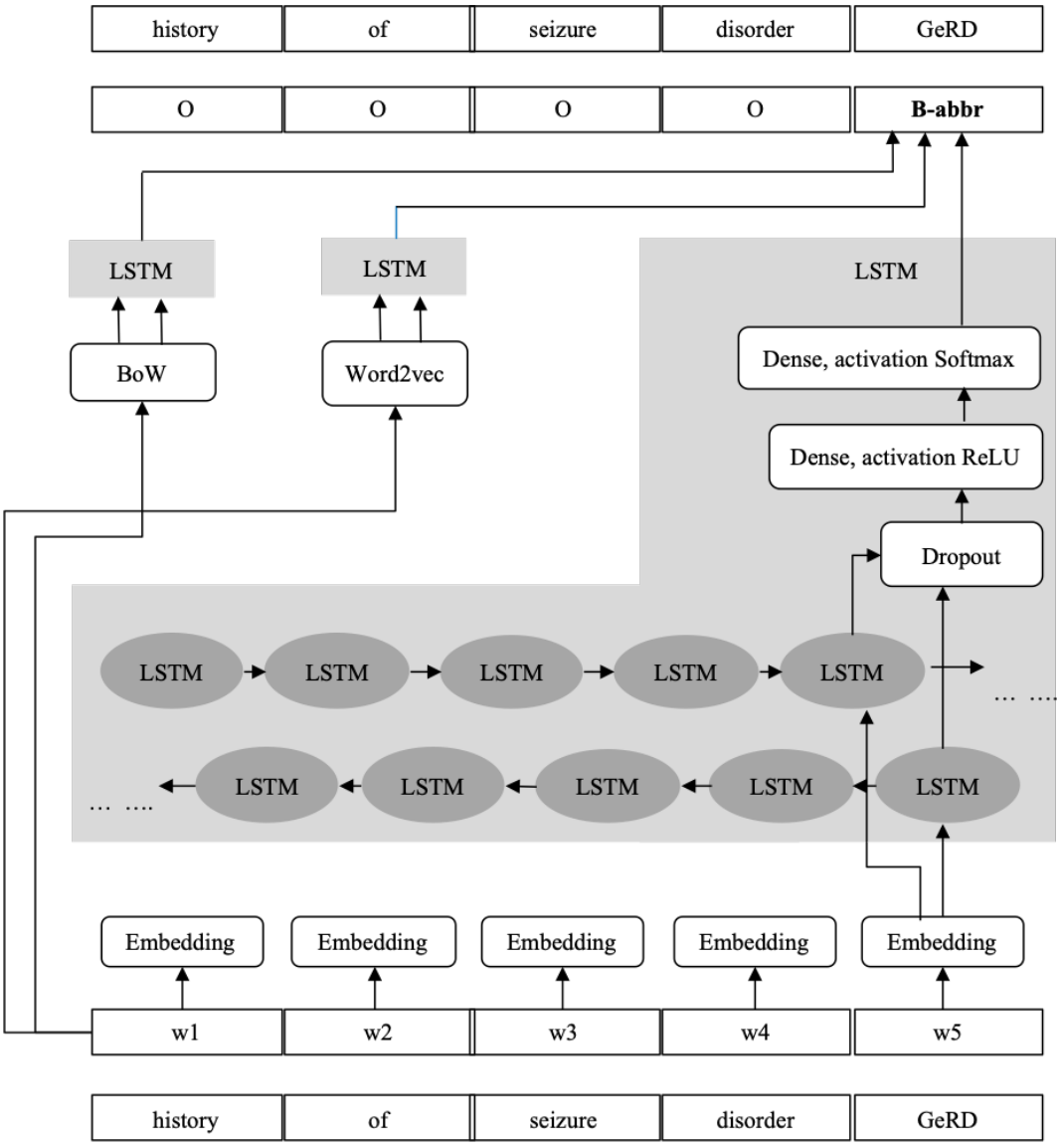

FIGURE 5.3: Design of model

pre-processing and processing components described in the methods section. As for pre-processing component, oversampling the dataset in the pre-processing part was verified to increase the performance of the informal abbreviation classifier. As for the processing component, it was verified that concatenating the word's representation such as BoW and word2vec in the embedding layer of the LSTM model could increase the performance of the informal abbreviation classifier.

The precision of the classifier when the abbreviation samples are added was evaluated against the basic model. The precision when the word representation inputs, such as BoW and word2vec, are added to the model as multiple feature inputs was also evaluated.

5.4.1 Data set

Our data were acquired from a publicly available medical notes corpus [106] in English. The data consist of medical notes of varied types from different medical specialties, such as progress notes, SOAP notes and discharge summaries [107]. This corpus is quite general, and contains informal abbreviations, including various writing styles, such as upper and lower case only and mixed cases, alphanumerics, and alphanumerics with text symbols.

Table 5.1 shows the distribution of abbreviations and non-abbreviation words in the dataset. A total of 475 sentences were extracted from randomly selected medical notes. The sentences represent most of the informal abbreviations for our case study. Using the pre-processing strategies, 7159 samples, 704 of abbreviations (10%) and 6455 of non-abbreviations (90%), were acquired. Furthermore, the vocabulary extracted from the data set consisted of 1629 unique words, including abbreviations.

Each sample consists of a sequence of five words. This sequence length was used because classifying longer sequences is more difficult [109]. The 7159 5-word samples were split into a training dataset (80%) and a test dataset (20%). Both datasets had the same proportion of abbreviations. This ratio could be specified as a parameter of the sample generator.

TABLE 5.1: Data set distribution

Free text data	Size	Example
Sentences	475 sentences	"coronary artery disease history of seizure disorder GERD bipolar"
Unique vocabulary	1629 vocabs	PSA, GERD, mg, TEST, Cardiolite, cm, Dr, STRESS
Abbreviation words	279 words	Q-fever, h/o, mg, G3P3, PSA, PMH, DVT, ml, cm, Dr
Non-abbreviation words	8786 words	TEST, STRESS, incidental, DATA, Cardiolite
5-word abbreviation samples	704 samples	"of seizure disorder GERD bipolar"
5-word non-abbreviation samples	6455 samples	"coronary artery disease history of"

5.4.2 Implementation of the model

A bidirectional LSTM was used with forward and backward size 32, dropout of 0.3, and two dense layers, the first one with size 32 and ReLU [108] activation (to preserve the input's sparsity), and the last one with activation softmax [109] for the final prediction with size 3. A batch size of 32 and 200 epochs were used for training.

From the same basic model, several combinations of inputs were implemented and evaluated. In the first test, only the word indices converted into embedding vectors with

size 100 was used. For the remaining tests, the combinations of: pre-trained BoW only; pre-trained word2vec only; each of these vectors concatenated to the embedding vectors; a combination of BoW and word2vec; and all vectors together was evaluated. The BoW matrix had a size of 1629 (same as the size of the vocabulary), and the word2vec matrix had a size of 100.

In the processing part, modification of the input structures with embedding replacement and concatenation between embedding, BoW, and word2vec was done. The remaining LSTM layout such as size of layers, dropout, dense and activation layers were the same for all input combinations. The concatenation between embedding vectors and additional matrices uses LSTM with the same configuration.

5.4.3 Evaluation method

The performance of our classifier was evaluated, by analysis from the perspective of both the pre-processing and processing components. From the pre-processing perspective, the model's performance was evaluated based on the abbreviation samples added to the pre-processing part of our methods. It was hypothesized that a more balanced sample increases the classifier's performance. From the model's processing perspective, the model's performance was evaluated based on all the concatenation combinations of the input matrices from the embedding layer of LSTM, pre-trained BoW matrix, and pre-trained word2vec matrix. Concatenation of inputted matrices was hypothesized to increase the classifier's performance.

As the original dataset suffers from class imbalance, evaluation of the data sampling is necessary to understand the effects of increasing the number of samples. The informal abbreviation is a special case of Out-of-Vocabulary (OOV) word, which requires more features to differentiate it from the ordinary words. Therefore, input combinations were evaluated to assess the effect of input combinations in the model.

Increasing the model's precision was the focus in this experiment since the cost of false positives is high. Due to the number of negative (non-abbreviation) samples being very large, the model has low precision. This will result in many words being classified as abbreviations, i.e. many false positives. When the rate of false positives is too high, the system is no longer reliable for detection. Precision is more related to the positive class (abbreviation class) than to the negative class (non-abbreviation class) since it measures the probability of correct detection of the positive class.

In correlation with the interface design of the support system, by focusing on increasing the classifier's precision, incidences of a non-problematic text such as a non-abbreviation

word that is falsely detected as a problematic text or abbreviation word (false positive) is minimized, reducing false notifications appearing in the user interface (UI) to the physicians.

Using small window size samples, such as 5-word samples, the recall and F1-score are not as high as the precision because the effect of the number of words that are analyzed in the context of an abbreviation. When the window size is large, recall increases because words farther away from the abbreviation are taken into account and more abbreviations are likely to be found. The precision will be higher when the window size is small [110].

5.5 Results

The basic model, which is the baseline, had a precision of 68.6%. Only embedding was used as input. The original dataset contains 10% of the abbreviation samples, which represents only 2% of all the abbreviations. That might explain the model's low performance.

A sample generator was used to increase the precision of the baseline model by adding more abbreviation samples to the original dataset in increments of 10% until an intended precision was reached. The enhanced data set contained 90% of abbreviation samples, representing 18% of abbreviation words in total. Table 5.2 lists the means and standard deviations of the precision, recall, and F1-score from five trials at various levels of abbreviation samples. The precision improved as the ratio of abbreviation samples increased. The recall and F1-score were greatest when the level of abbreviation samples was 40%, reaching 55.1% and 65.2% respectively. When the abbreviation samples were at 90% of the total samples, the precision was 91.4%, recall was 48.7% and the F1-score was 63.3%. The precision steady increased until its highest precision at 90% of abbreviation samples, while the recall and F1-score fluctuated. The new dataset was used as a baseline for the next experiment using additional inputted matrices.

Table 5.2: Improvement using samples generator

Abbr samples	Precision (Avg)	(SD)	Recall (Avg)	(SD)	F1-score (Avg)	(SD)
10 percent (baseline)	68.6%	14.1	38.2%	0.3	48.6%	4.1
20 percent	70.6%	15.1	46.1%	2.4	55.0%	4.6
30 percent	77.1%	7.9	42.7%	5.8	54.5%	3.0
40 percent	80.4%	6.1	55.1%	1.7	65.2%	0.8
50 percent	83.2%	4.2	48.3%	3.3	57.3%	3.3
60 percent	85.0%	5.7	49.5%	2.7	62.4%	2.1
70 percent	85.1%	5.9	49.5%	2.8	62.4%	1.2
80 percent	91.1%	8.6	45.5%	4.9	60.3%	3.8
90 percent	91.4%	1.7	48.7%	6.2	63.3%	5.2

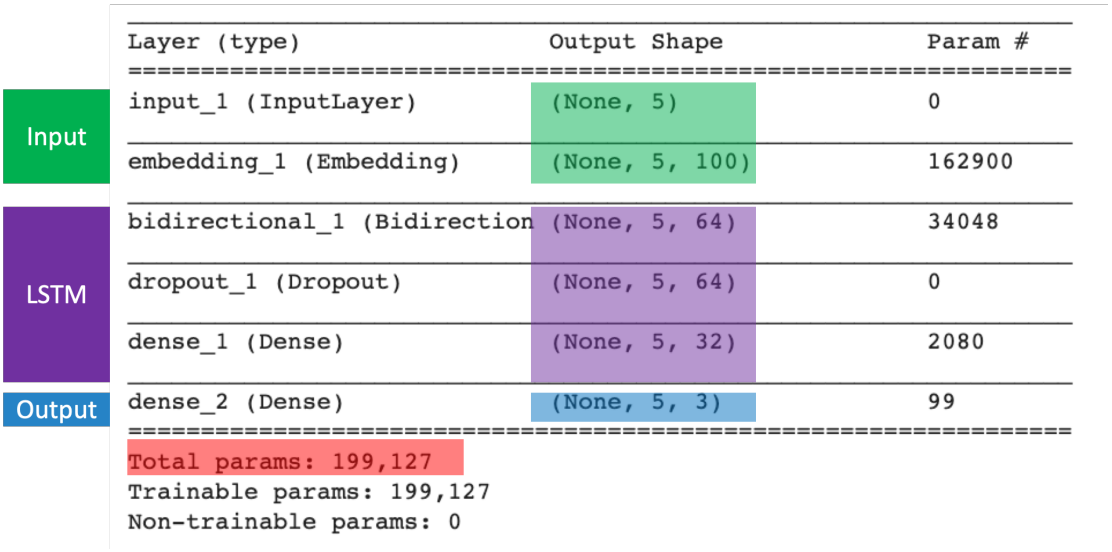

FIGURE 5.4: Model with embedding (baseline model)

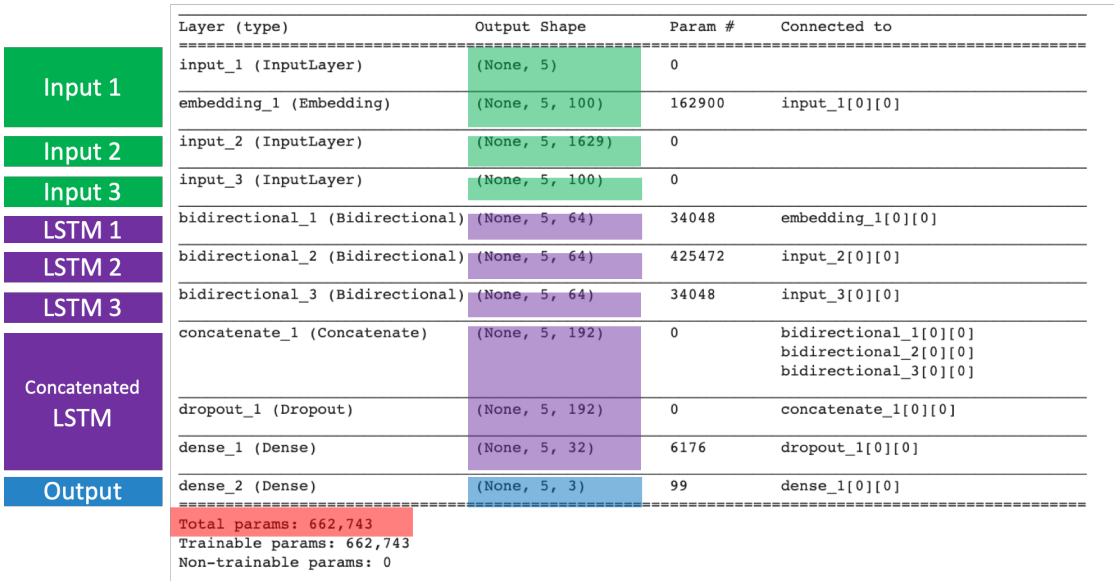

FIGURE 5.5: Model with embedding and BoW and word2vec

Table 5.3 shows the precision improvement when the embedding is replaced with the BoW matrix, concatenation of embedding with BoW matrix, concatenation of BoW matrix with word2vec matrix, and concatenation of embedding, BoW matrix and word2vec matrix (Appendix C). Figure 5.4 shows the model with embedding input as the baseline model and Fig. 5.5 shows the model with the highest performance with three inputs, which are embedding, BoW, and word2vec.

The recall and F1-score increased when the embedding was replaced with the word2vec matrix, concatenation of BoW matrix with word2vec matrix, and concatenation of embedding, BoW matrix, and word2vec matrix. For the concatenation of embedding with BoW matrix, the recall slightly increased, while the F1-score did not change.

TABLE 5.3: Improvement using additional matrices

Inputs	Precision (Avg)	(SD)	Recall (Avg)	(SD)	F1-score (Avg)	(SD)
Embedding (baseline)	91.4%	1.7	48.7%	6.2	63.3%	5.2
Word2vec	74.8%	2.6	55.8%	2.7	63.8%	1.2
BoW	91.6%	2.4	43.2%	1.5	58.7%	1.3
Embedding+Word2vec	84.7%	5.7	45.6%	1.7	59.2%	1.5
BoW+Word2vec	92.0%	4.6	53.1%	3.4	67.2%	2.6
Embedding+BoW	92.6%	8.0	48.8%	6.6	63.3%	3.7
Embedding+BoW+ Word2vec	93.6%	2.7	57.6%	8.0	68.9%	5.5

When the level of abbreviation samples was 90%, the BoW matrix increased the precision of the baseline model to 91.6%, the BoW matrix and word2vec matrix concatenation increased it to 92.0%, addition of BoW matrix to the embedding of baseline model increased it to 92.6%, and concatenation of embedding, BoW matrix, and word2vec matrix increased it to 93.6%. Recall was increased to 55.8% by the word2vec matrix, to 53.1% by concatenation of BoW and word2vec, very slightly to 48.8% by concatenation of embedding and BoW, and to the maximum of 57.6% by concatenation embedding, BoW matrix, and word2vec matrix. The F1-score was increased by word2vec to 63.8%, by concatenation of BoW and word2vec to 67.2%, and to its highest at 68.9% by concatenation embedding, BoW matrix, and word2vec matrix.

The results showed that the LSTM based model could accurately detect abbreviations made in diverse writing styles in free text medical notes. In addition, our model was able to predict abbreviations even when there were few in the dataset.

The model's ability to predict the abbreviations relies on the fact that it learns from the context in which words appear in sequences [111]. Most medical notes are written following a certain narrative, which creates a relevant context in the sequence of words [112]. Therefore, although some words are informal abbreviations, the LSTM is able to recognize them by learning from surrounding words [113].

The precision, recall and F1-score of the model can be increased by addition of abbreviation samples and concatenating inputted matrices. In our case, the precision of the classifier was the metric under the spotlight because it is important for information extraction tasks that often demand a high precision [114]. The recall and F1-score are not as high as the precision, and it is common for the information extraction tasks with small window size samples, such as 5-word samples [110].

5.6 Discussion

The effects of sample population on the model's precision are discussed from the pre-processing perspective. The effects of additional pre-trained features on the model's precision are discussed from the processing perspective.

In our analysis, the LSTM baseline model's precision was increased by adding more abbreviation samples, as shown in Table 5.2. With every 10% increase in abbreviation samples, the precision increased by between 0.3 and 6.5 points. The smallest precision increment occurred when going from 80% to 90% of abbreviation samples, while the largest increment occurred when going from 20% to 30% of abbreviation samples. Usually, LSTM has a good performance in large datasets with little class imbalance [115]. By combining a sliding window and a sample generator, it was possible to increase the number of samples while reducing the class imbalance, which allowed for an increase in the model's precision.

It was also possible to increase the LSTM baseline model's precision by including additional pre-trained features as a complementary input, as shown in Table 5.3. The addition of BoW to the model's input increases the precision by between 0.2 and 2.2 points. The addition of word2vec increases the model's precision if it also includes the BoW matrix addition. This takes advantage of the fact that the LSTM model can process multiple features for its learning [99]. Initially, the basic input – embedding vectors – is used to differentiate between the classes in the binary classification (abbreviation vs non-abbreviation). As additional features are added to the LSTM model, it can improve its ability to differentiate between them. For this reason, word frequency (BoW) and word2vec vectors are commonly used features in natural language processing [116].

Adding BoW increased the model's precision. When using a pre-trained BoW instead of word2vec, an even higher performance was achieved. This result shows that the LSTM model for the detection of informal abbreviations is better at learning the information about word frequency represented by the BoW matrix than by the word2vec matrix. This is due to the small number of abbreviations in our dataset, which enables the model to distinguish low-frequency abbreviations from high-frequency non-abbreviations words. Compared with word2vec, the small number of abbreviations renders the model difficult to differentiate between abbreviations and non-abbreviations due to lack of context available from the corpus dataset.

Interoperability standards, such as SNOMED CT, state that medical terms should be expressed in a structured and unambiguous way, providing a semantic expression whose meaning can be agreed upon by different systems, in a consistent and clearly expressed manner. However, in the case of free text medical notes, medical expressions are some-

times written using informal abbreviations that may not be detected by SNOMED CT based processors. This could negatively impact patient care and generate a burden for third party usage [117].

Necessary from the coders' perspective, abbreviation can be detected with high precision using LSTM-based model, as shown in Table 5.2 and Table 5.3. Additionally, our approach expands the field of application of the LSTM-based model since the technique has never been used before for this kind of task.

An optimal classifier, which is classifier with high precision, for handling informal abbreviations can be extended to the detection from the coders' perspective, such as helpful keywords and incomplete structured data, as described in Section 4.3. It can be extended because most of the coders' perspective is about entity recognition from free text, which can be handled using an LSTM-based NER approach such as the informal abbreviation recognition model described here.

For an optimal classifier, an existing imbalance class issue can be solved using multiple input feeds into the model to increase an underrepresented class, as discussed in Section 5.5.

For detection based on the coders' perspective, such as incomplete data detection, detection of SOAP note parts, helpful keyword detection, and indicator of SOAP note problems, the informal abbreviation detection model can be expanded.

First, for incomplete data detection, each sentence in the SOAP notes was labeled as either as incomplete or complete. A similar binary classification to classify incomplete and complete sentence was used. The difference between incomplete data or incomplete sentence detection and informal abbreviation detection is that in the former, sentence level labelling [118] was used, while in the latter word level labeling in the form of a training dataset was used. When input into an LSTM or other deep learning model, at the word level labelling, the word transforms into vector space using word embedding such as Word2vec. Similarly, sentence embedding embeds a full sentence into a vector space using sentence2vec [119]. In practice, sentence embedding might look like this: "the patient has fever" becomes [0.2 ; 0.1 ; -0.3 ; 0.9 ; . . .]. The sentence embedding retains some acceptable properties, as it inherits features from its underlying word embedding. Thus, sentence embedding might be used for varying purposes, such as to detect incompleteness of structured data from the sentence. After sentence embedding takes place, the embedding could be input into the LSTM model to classify sentences as to whether it has complete or incomplete structure, based on the coders' perspective. The flow from inputting sentence embedding into the LSTM model until outputting an incompleteness label is the same as in our informal abbreviation binary classification.

Second, for detection of SOAP note parts, each paragraph in the SOAP notes was labeled S, O, A or P. Each paragraph was transformed into a vector space using paragraph2vec [120]. The only difference from word2vec is the inclusion of documents or SOAP notes along with words as input nodes. The rationale behind including SOAP notes as input nodes is based upon considering SOAP notes as another source of context. After paragraph embedding is produced, the embedding can be inputted into the LSTM model to classify whether the paragraph is part S, O, A or P. The flow procedure from inputting paragraph embedding to the LSTM model until outputting part label is the same as in our informal abbreviation binary classification.

Third, for helpful keyword detection, each phrase and each related abbreviation in the SOAP notes was labeled as a helpful keyword. A phrase is a group of words that express a concept. A phrase is classified as a helpful keyword if there is also a related abbreviation detected in the SOAP notes, for example "petechia on skin" is detected as being a helpful keyword if the abbreviation "df" is also detected. To a transform phrase into vector space or a phrase embedding, phrase2vec [121] was used. After phrase embedding is produced, the phrase embedding can be inputted into the LSTM model along with the abbreviation embedding to classify whether the phrase is a helpful keyword or a non-helpful keyword. The flow procedure from the inputting of the phrase embedding and abbreviation embedding into the LSTM model until the outputting of the helpful keyword's label is same as in our informal abbreviation binary classification. In this helpful keyword detection model, the phrase embedding and abbreviation word embedding are two combined inputs that feed into the model. The model learns from the combined inputs to classify the phrase as a helpful keyword or non-helpful keyword.

Fourth, for indication of SOAP note problems, the system aggregates the outputted labels from the abbreviation classifier and the incomplete data classifier to quantify the intensity of SOAP note problems.

It was believed that all classifiers and indicators based on the coders' perspective have a higher accuracy if there is no issues of imbalance class when training the model. Although it is common to have class imbalance issues in the Natural Language Processing (NLP) domain, a highly accurate classifier can be achieved via multiple inputs, such as BoW and pre-trained word2vec fed into the model and by increasing underrepresented classes using sliding windows and a sample generator.

5.7 Summary

It is not acceptable that only the physician who writes medical notes is able to understand them. Multiple people who participate in the medical care process must be able to understand them and agree on the meaning of all terms. In this sense, if informal abbreviations such as those shown in Table 5.1 can be clarified in a disambiguation process, higher degrees of semantic interoperability can be achieved. This allows relevant clinical information to be recorded using consistent, common representations during a consultation and supports the sharing of appropriate information with others involved in delivering care to the patient.

This study had the following limitations. First, the dataset was composed of random samples from a public database of medical notes, which may not fully represent all the

writing styles found in authentic medical notes. To further validate and improve the results, it is necessary to use larger datasets with medical notes from different hospitals and clinical settings. Additionally, the dataset itself was small in comparison to those used in other studies on natural language processing. However, in the specific context of medical notes, this is to be expected as there would be issues of privacy and confidentiality. In fact, most studies can only have access to a limited number of authentic medical notes. In previous studies, mtsamples.com dataset was used as a clinical study benchmark for extracting family history diagnosis from clinical texts [122] and for identifying representations of drug use [123]. In addition, for our evaluations of the classifier to detect informal abbreviations across many categories with class imbalance problems, such as the number of abbreviations beings far less than the number of non-abbreviations, a public dataset such as mtsamples.com fit was appropriate for our study. It is arguable that if the classifier performs highly in detecting abbreviations of low frequency, then it should perform better in detecting a higher population of abbreviations that would be commonly found in a real dataset.

This study can be extended to detect multiple undefined medical entities from free text medical notes, such as symbols, for example "(+)", "(-)", and "(↑)", with diverse variation of writing styles similar to the informal abbreviations cases. Symbols are a kind of abbreviated version of common words that are often used in the medical notes to express the patient's condition, such as "(+)" for "positive", "(-)" for "negative", and "(↑)" for "increased".

Using the LSTM based model to recognize informal abbreviations in diverse writing styles that are commonly found in free text medical notes is innovative. Our approach

differs from previous approaches that only focus on the formal abbreviations with predefined features [92]. Using our approach, the informal abbreviation detection had a precision of 93.6%, recall of 57.6%, and F1-score of 68.9%, for a small population dataset. Thus, our study may serve as a case study in future NER research using small datasets. The precision was used as an evaluation metric to minimize false positives and optimize the specificity of the classifier. However other metrics such as recall and the F1-score can be still be used if necessary.

For possible future extension of our work, it will be useful to develop a writing support systems, which can recognize informal abbreviations in real-time while the physician is typing it and raise appropriate indicators for confirmation of the meaning of informal abbreviations, and thus increase the semantic interoperability of the EMR system. However, this would not be suitable physicians' natural writing procedures if it were to introduces unintended text by not being 100% accurate in its predictions. Although the LSTM-based informal abbreviation detection achieved a high precision (above 90% precision), it might still burden the physicians in some cases. In order to develop a writing support system which takes into account the coders' perspective, such as converting into structured data informal abbreviations that physicians might use in their normal writing, a support system user interface (UI) has been designed and its function evaluated. The design and evaluation of the UI are the first steps to developing a real working system from the perspective of user's goals. They are described in Chapter 6.

Chapter 6

Design and Evaluation of the Interface of the Support for Writing Medical Notes

The chapter presents the design and evaluation of the user interface of the proposed supported for writing medical notes. The design is compared with user interfaces commonly used for writing medical notes. in terms of its efficiency to write medical notes and in terms of its ability to capture more data required by coders. It includes results of a preliminary quantitative evaluation in which the proposed user interface was compared with a user interface commonly used for writing standard medical notes. Part of this chapter was published in HAI conference (Appendix F).

6.1 Overview

The coders' perspective on non-standard abbreviations and incomplete medical data in the medical notes should be presented to physicians in order to get their feedback about the issues. The interaction mimics the coders' task asking physicians for elucidation of problematic SOAP notes. The proposed system presents suggestions to improve the interpretability of the finished medical notes, for example the system gives advice to add more data about vital signs, or to change non-standard abbreviations to standard ones. In the writing process, there are three sub-processes, namely planning, translating and reviewing/revising the text [124], as shown in Fig. 6.1. The UI was designed to support physicians when reviewing their medical notes. The review part of the writing process is an intersection between the physician's writing process and the coders' data requirements. When reviewing, the physician may be reminded by notifications that

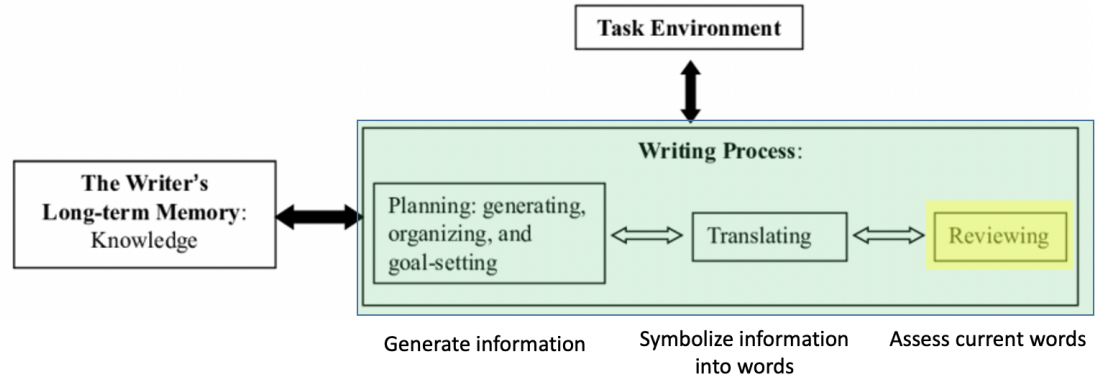

FIGURE 6.1: A cognitive process theory of writing [124]

certain current words or abbreviations can be problematic to the coder. In the current design, the simplest form of notification was used, which present only when the physician has finished writing the notes. However, the design could easily be modified to give notifications in real time.

Several "auto" or automatic approaches, e.g. auto-completion and auto-correction, can render text easier for the coder to interpret. They work best in domains with a limited number of possible words, such as with command line interpreters, when source code editors are used to write structured and predictable text, or when some words are very common (such as in e-mails).

In clinical settings, physicians write medical notes in a wide variety of ways and frequently use non-standard abbreviations or informal writing styles. For many, the introduction of an "auto" approach which is intended to improve their medical notes is seen as a burden. Worryingly, auto approach has resulted in unintended prescriptions, improper dose/quantity and incorrect preparation of medication [125]. The auto approach may also give unintended or incorrect data requirements from the coders' perspective.

In order to support the physician's natural way of writing notes, while meeting the coders' requirements, an adviser or reminder approach was proposed. In this approach, the physicians have full degree of freedom in the way they write. The system presents the coders' perspective as advice or suggestions to the physician via the user interface.

The proposed UI was evaluated by comparing it with UI commonly used to write medical notes. First, was a comparison of how well physicians were able to write their notes, and secondly how well it captured more of the coders' required data.

To evaluate improvement of problematic SOAP notes, the following procedure was used. First, a coder analyzed the notes and made comments about the problems. The total number of comments was registered as the initial score of the notes. Then the physician

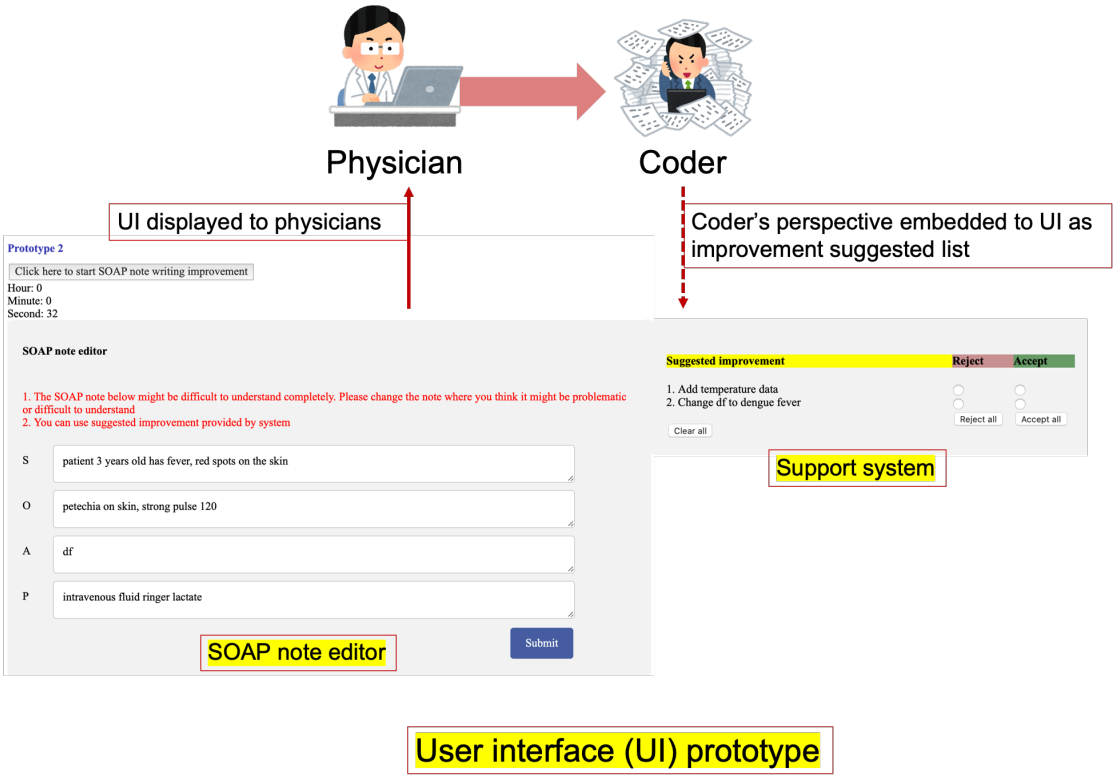

FIGURE 6.2: Proposed user interface

was asked to correct the notes using the system. Two versions of the system can be used, one that offers support for correction and another that presents just a group of text fields that the physician can freely edit. After the physician has completed that task, the number of changes made by the physician (words or phrases changed) is subtracted from the initial score to generate the final score. The smaller the final score, the better it is. For example, if the initial score is 10 and the physician adds or changes 7 words or phrases, the improved score is 3. A score of 0 means that the SOAP notes are no longer problematic for the coder because all problematic comments have been addressed by the physician.

Figure 6.2 shows our proposed user interface design that was embedded into a typical SOAP note editor.

6.2 Experiment

Ten problematic SOAP notes in the Indonesian language were used to create 10 pairs of web pages (Appendix D), half with and half without a support system. Figure 6.3 shows a pair of web pages from a SOAP note. The physicians were asked to review and improve SOAP notes so they are less problematic for the coder. When there is a support

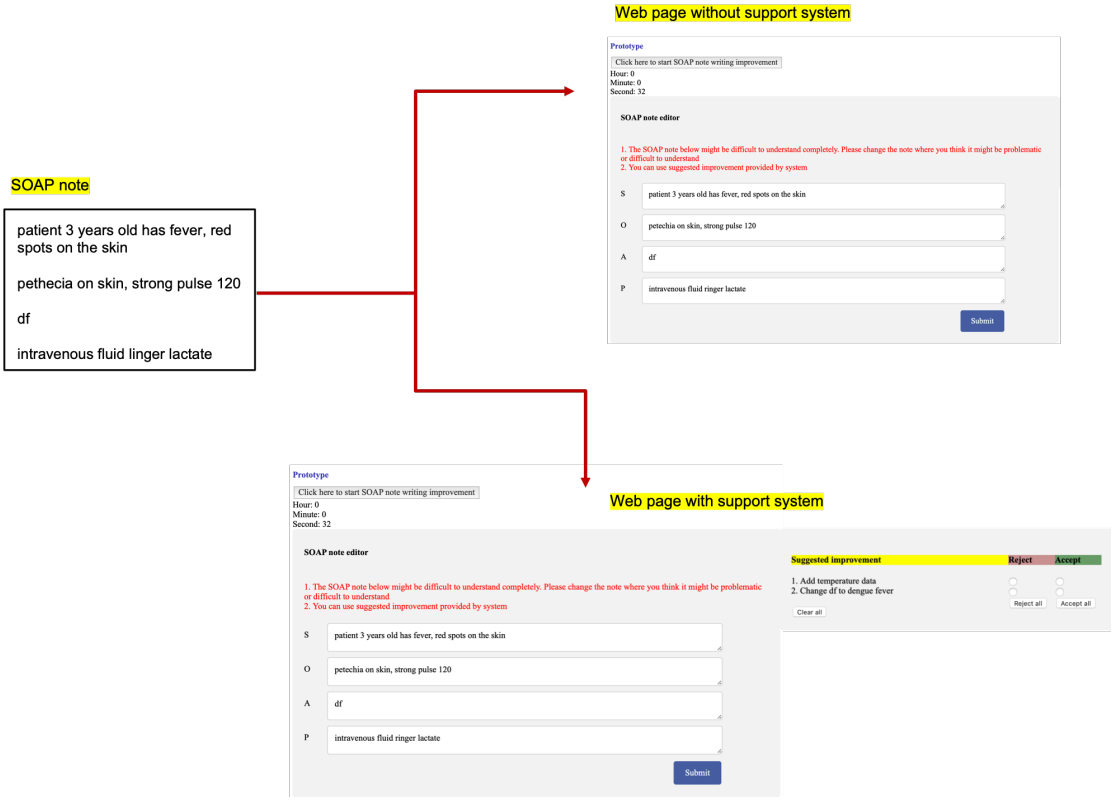

FIGURE 6.3: A paired web pages from one SOAP note

system, the physician can make improvements by accepting suggestions appearing in the Suggestion List Frame to the right of the SOAP note Editor Frame, or by simply editing the notes in the Editor Frame. When there is no support system, the physician can make improvements only by editing the notes in the Editor Frame. On clicking the Submission button, all changes were saved. The task was timed by a built-in counter.

6.2.1 Experimental procedure

The suggestion list was not displayed by the system's algorithm but by their manual entry into the web pages as a predefined suggestion list. The Wizard of Oz method was used by manual entry of the suggestion list as the computer's response via UI. The Wizard of Oz method is a process that allows a user to interact with an interface without knowing that responses are being generated by a human rather than by a computer. Someone behind-the-scenes ho is acting as a wizard and mimics the system's algorithm output [126]. The Wizard of Oz was used to test user's response to the system's UI at the very beginning of the system development [127]. Creating a Wizard-of-Oz prototype starts with determining what the experimenter wants to test or explore. Then the experimenter needs to figure out how to fake the functionality needed to give the user a realistic experience from their viewpoint. For example, the experimenter could prototype

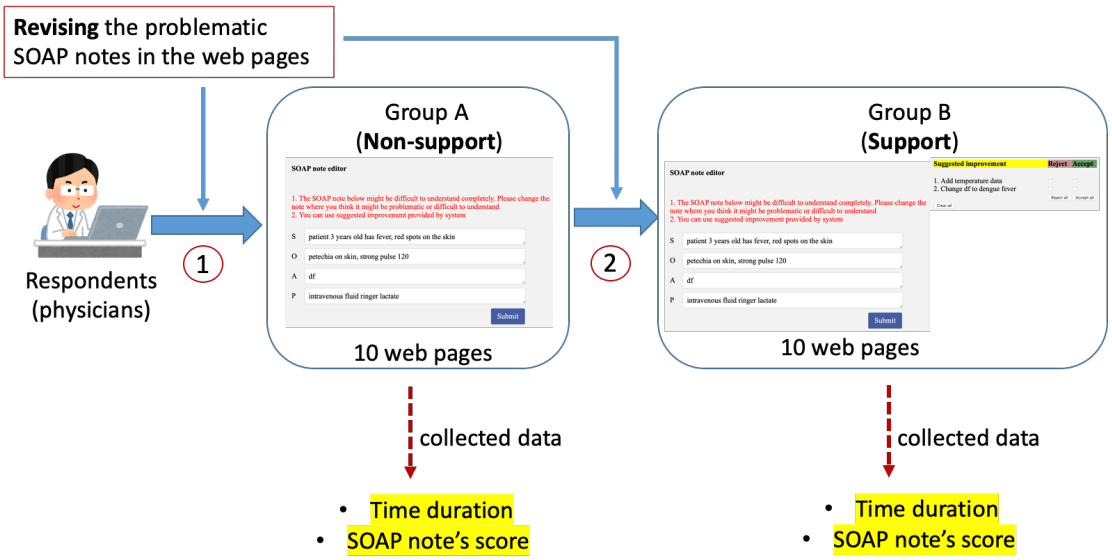

FIGURE 6.4: Experimental procedure

a jukebox without creating the mechanics and use a hidden person to play the selected songs to the customer [128].

The two Indonesian physicians (two female physicians) were recruited to test the system. Each physician were asked to review the 10 SOAP notes without the support (Group A notes) and then the 10 SOAP notes that had embedded support (Group B notes), as shown 6.4.

Changes and additions to the SOAP notes and the duration of the task were noted. The Group A and B notes scores were calculated by I = O – C, where I and O are the improved and original scores and C is the number of changes or additions.

6.2.2 Evaluation method

The paired t-test [129, 130] was used to compare Groups A and B for differences in the task duration and scores as illustrated in Figure 6.5.

6.3 Results

The scores and times for each SOAP note and the means of each set of 10 for each physician are given in Table 6.1. Figs. 6.6 and 6.7 show the differences in means and standard deviations of the scores task durations between Groups A and B.

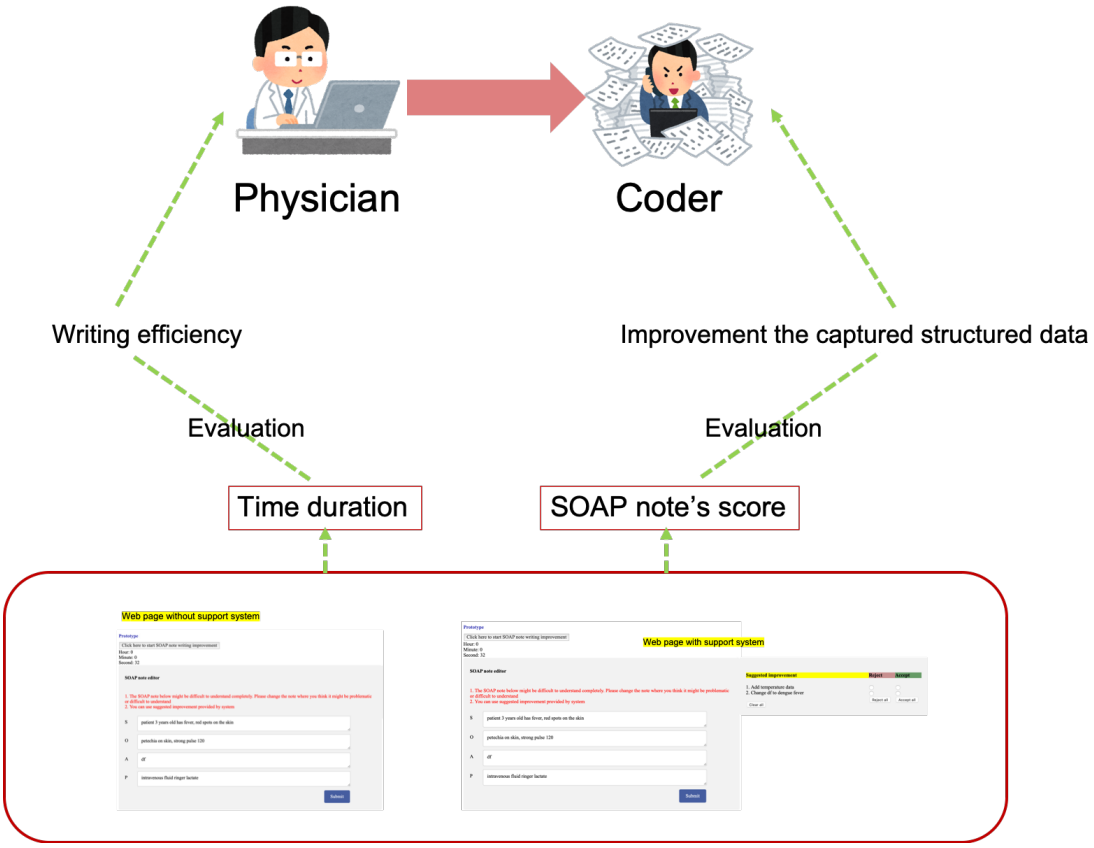

FIGURE 6.5: Factors to be evaluated

The differences in time and score between Groups A and B for each SOAP note are shown in Figs. 6.8 and 6.9, respectively, with yellow bars indicating improvement. Table 6.2 and 6.3 show the paired t-test results.

6.4 Discussion

As seen in Figs. 6.8 and 6.9, the score, and hence the writing efficiency, increased in 16 out of 20 (80%) and more of the data required by coders was obtained in 17 out of 20 (85%) when a support system was provided. Furthermore, in 14 out of 20 (70%), there was an improvement in both of these measures.

Comparison of the two groups in 6.1, revealed an improvement in time efficiency and score with introduction of the writing support. Thus, the physicians were able to more efficiently review SOAP notes, and more data required by coders could be collected from the notes.

Although only two physicians participated, its initial results are promising.

TABLE 6.1: Collected time duration (t) and SOAP note's scores (s) from two groups

		Grup A (Non-support)				Group B (Support)			
Subject ID	Note ID	t in sec	Mean t	score	Mean s	t in sec	Mean t	score	Mean s
MD 1	note 1	105	**109.2**	12	**12.9**	73	**86.2**	11	**11.7**
	note 2	154		12		97		10	
	note 3	172		16		60		11	
	note 4	99		5		68		2	
	note 5	108		17		97		15	
	note 6	96		8		105		10	
	note 7	96		17		119		16	
	note 8	102		12		64		12	
	note 9	71		16		76		16	
	note 10	89		14		103		14	
MD 2	note 1	134	**109.4**	9	**11.6**	87	**66.7**	6	**10.2**
	note 2	139		4		130		0	
	note 3	71		13		25		12	
	note 4	148		1		90		0	
	note 5	63		16		63		13	
	note 6	110		10		79		11	
	note 7	68		20		52		18	
	note 8	69		12		49		13	
	note 9	102		16		37		15	
	note 10	190		15		55		14	

TABLE 6.2: T-test result for time duration differences between groups

t	df	p-value	Confidence interval,95%
3.335	1	0.1855	[-92.3061, 158.0061]
Group	Size	Mean	Standard deviation
Group 1	2	109.3	0.1414
Group 2	2	76.45	13.7886

TABLE 6.3: T-test result for scores differences between groups

t	df	p-value	Confidence interval,95%
13	1	0.04887	[0.0294, 2.5706]
Group	Size	Mean	Standard deviation
Group 1	2	12.25	0.9192
Group 2	2	10.95	1.0607

In a real working application, the Wizard of Oz would be replaced by an LSTM-based classifier as discussed in Chapter 5. The LSTM-based classifier is able to detect all the coders' entities or structured data requirements, such as abbreviations and incomplete data from the medical notes. The suggestion list presented by an adviser-like support

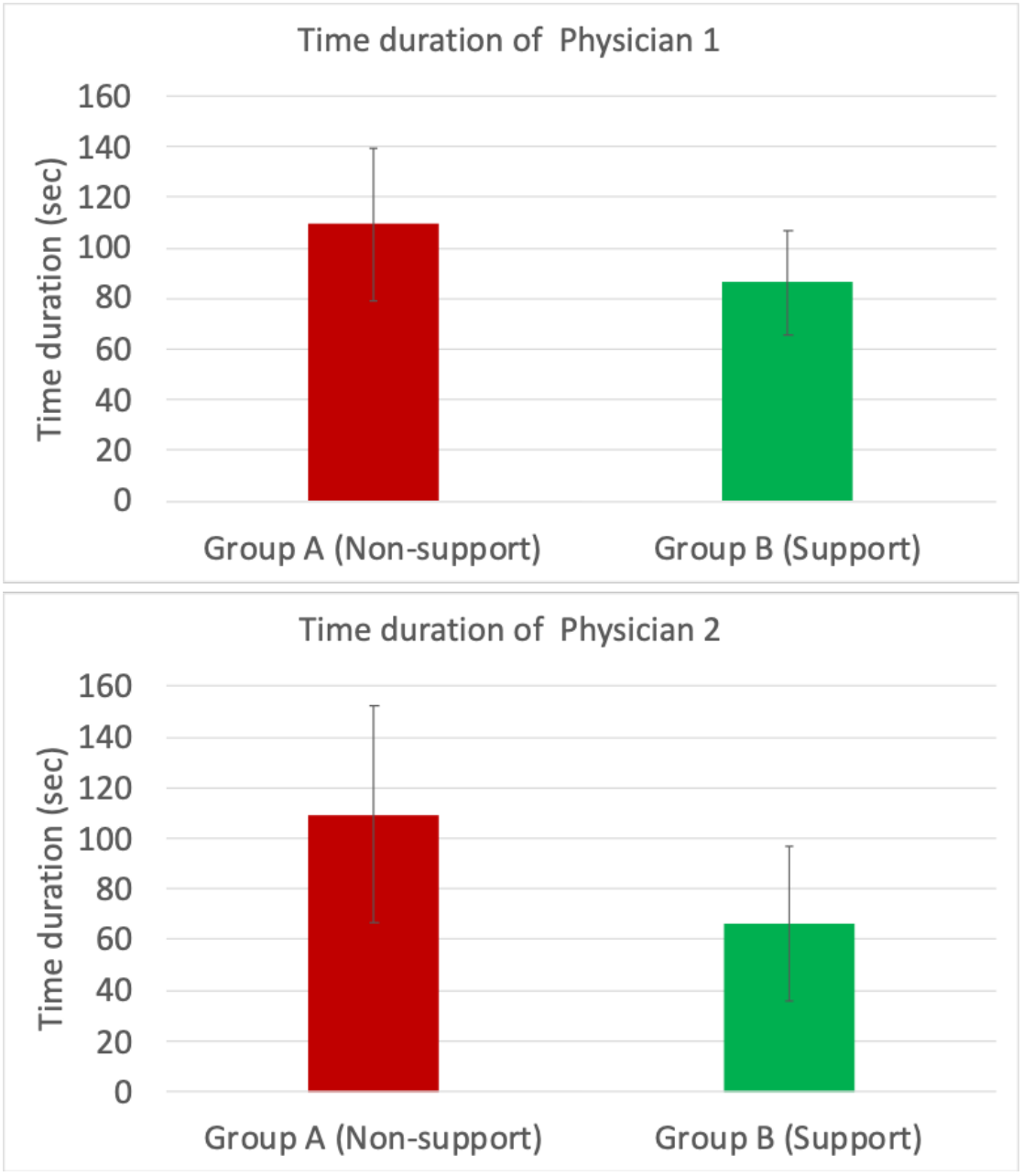

FIGURE 6.6: Time duration average and standard deviation between two groups

system could be extended to accommodate all possible coders' structured data requirements, such as unknown symbols. If there are other coders' structured data requirements based on the coders' perspective, such as unknown symbols, the support system would utilize an LSTM-based unknown symbols classifier. The detected unknown symbols would then be presented to the physicians via an adviser-like support system's UI. Using such a kind of modular mechanism, the additional LSTM-based entities detector is easier to be deployed and presented to physicians via the suggestion list frame in the support system's UI. Figure 6.10 shows the deployment design of LSTM-based entities classifier and support system's UI.

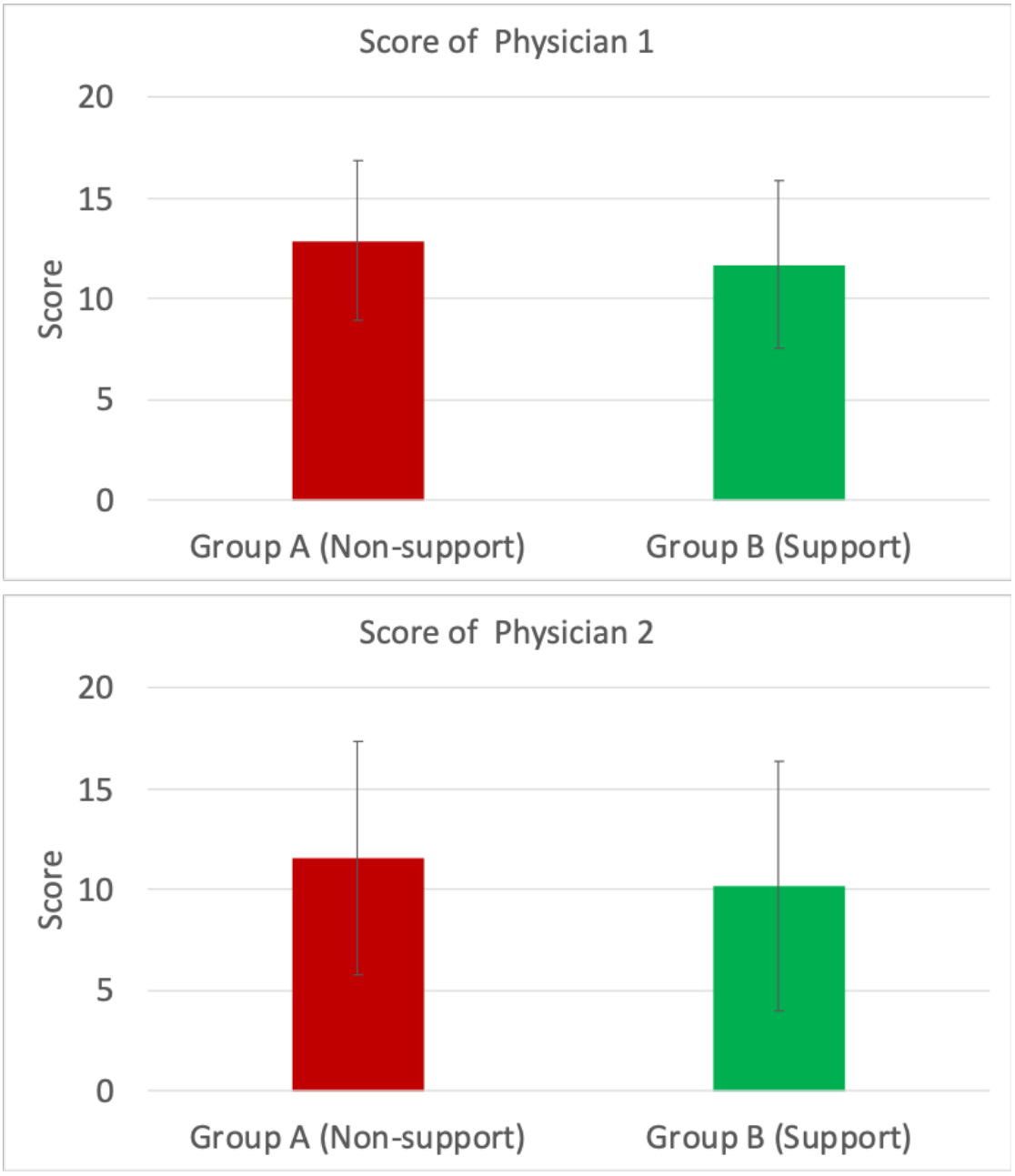

FIGURE 6.7: Score average and standard deviation between two groups

6.5 Summary

The writing efficiency was increased when the support system presented the physicians with a suggestion list. This confirms that physicians need to be guided or reminded to speed up their writing task. In addition, the reminder or adviser-like writing support performed well despite the suggestion list was not 100% relied upon, according to the physician's feedback (Appendix E).

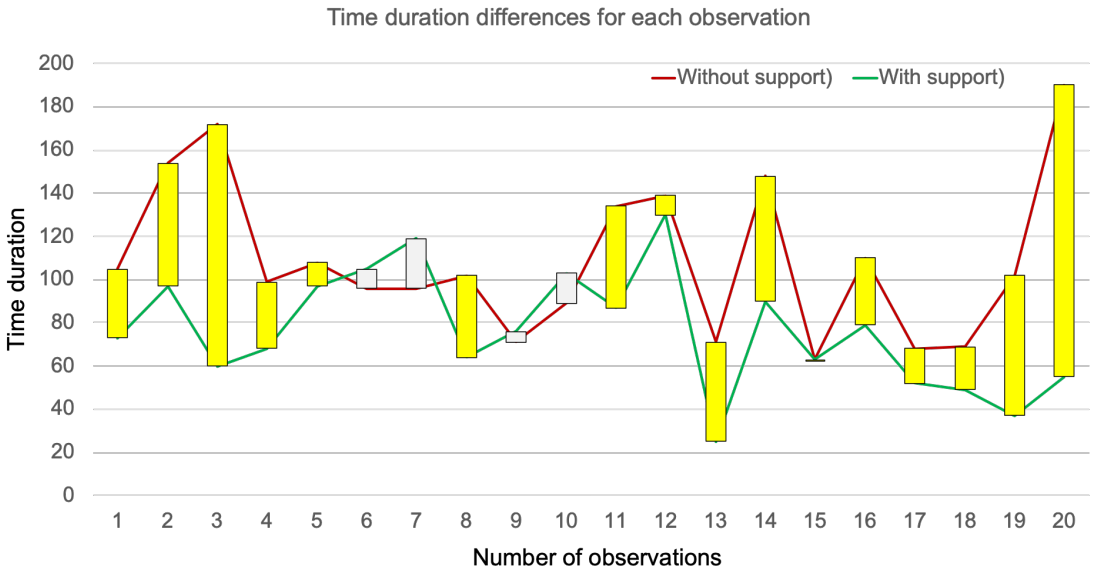

FIGURE 6.8: Time duration differences for each observation between two groups

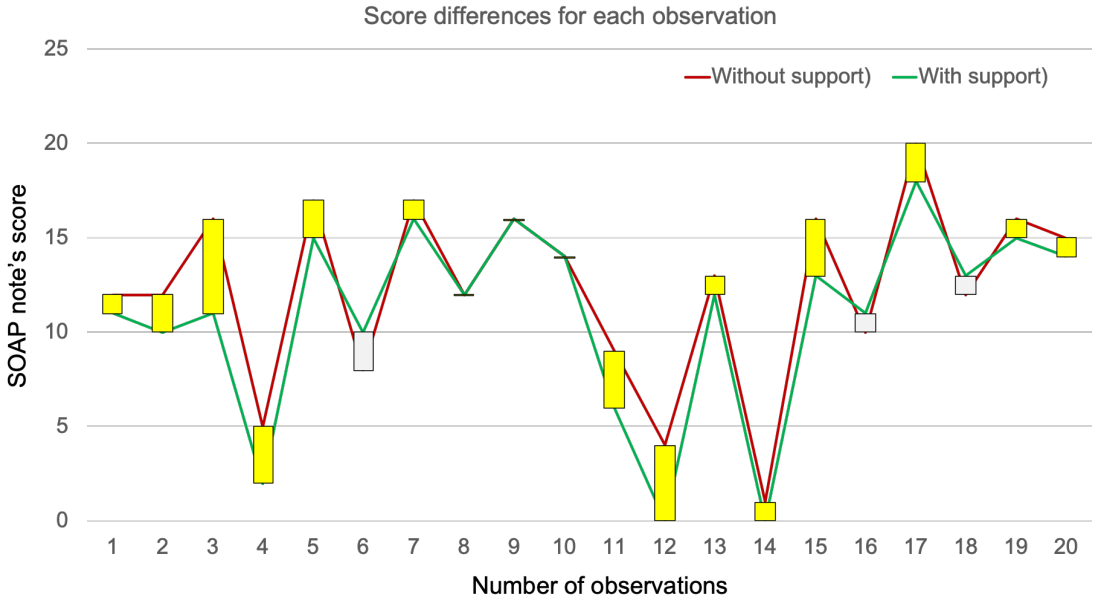

FIGURE 6.9: Score differences for each observation between two groups

The user interface is able to guide physicians to write SOAP notes with more data required by coders.

Writing with pen and paper or speaking into a microphone are easier than typing into a keyboard. However, in the era of EMR systems where all clinical data are expected to be in digital format and be structured and standardized to enable them to be processed optimally by a digital data processor, data entered in a less preferred manner e.g. a keyboard, may burden the physicians. A preferred natural writing manner is an important factor to consider when introducing new technology, like a writing support system.

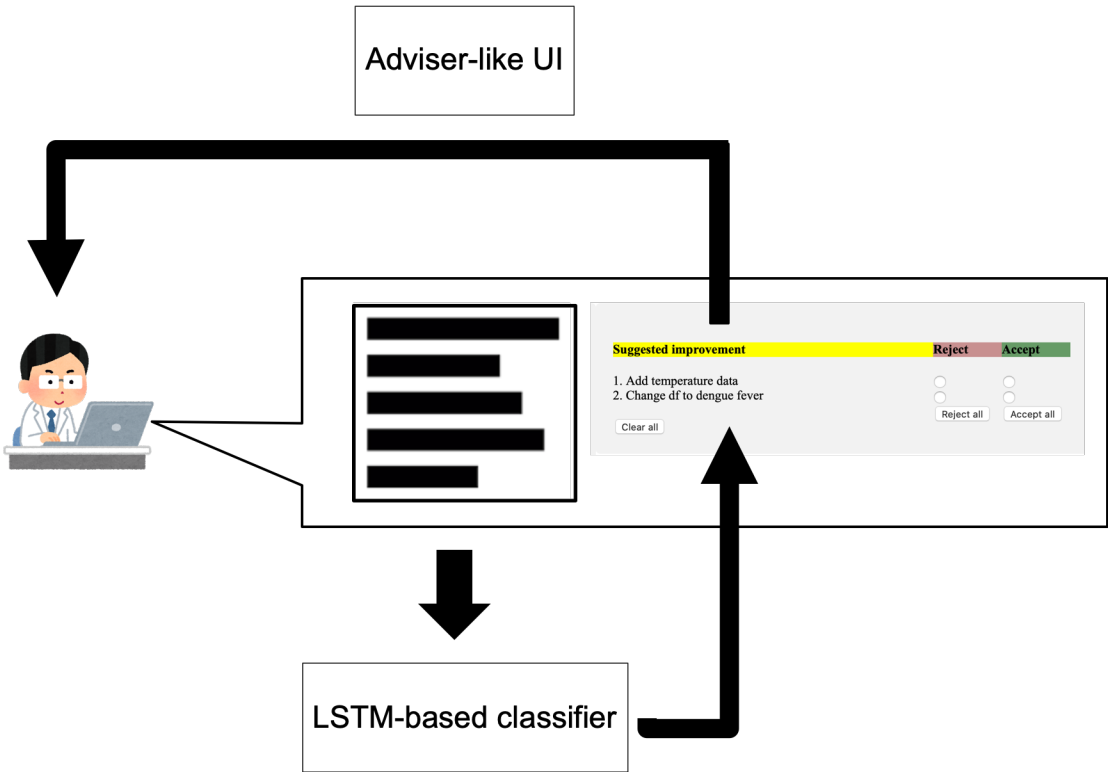

FIGURE 6.10: Deployment design of LSTM-based entities classifier and support system's UI

Several studies [131] showed that physician's top priority in considering new technology intended to support them is that the technology should not be a burden on their normal way of doing things. When a keyboard is used for data entry into an EMR system, with the physician's cognitive load already high, it is preferable that introduction of the support system should decrease rather than increase their cognitive load. Based on the physicians' evaluations of the design of the support system's UI, it was confirmed that introduction of this novel support system based on the coders' perspective integrated with their natural flow of documenting medical notes.

Cognitive load has a relationship with the effort expectancy, which predicts the acceptance of new technology by users. With the use of the novel support system based on coders' perspective being effortless, further development is worth doing because there is a high probability of it being accepted by users or physicians for their routine documentation tasks in the future.

The main player in medical notes documentation is the physician. However good quality medical notes documentation are also a benefit to other healthcare personnel including coders, clinical researchers and people dealing with primary and secondary usage of clinical data. Such personnel may be interested to note that new technology or EMR changes ought not hinder the physician's natural way of writing medical notes.

In evaluating the support system's UI, the support system's suggestion list was presented statically. It could be presented dynamically, in a new version. Although the static presentation showed a promising impact for physicians and coders alike, a dynamic suggestion list may have a similar or even better impact.

Chapter 7

Discussion

This chapter summarizes the findings and discusses their impact, generalizability and limitations.

This research introduced an innovative approach to making a support system for writing medical notes based on the data requester's perspective, particularly the coders' perspective. It presents the perspective as a support system for physicians writing medical notes in their natural way for an EMR system. The system improves the physician's writing efficiency. In addition, it provides more of the structured data required by coders. Furthermore, it provides effective support for human-to-human interaction. Figure 7.1 shows our approach of specifying the system's features based on the coders' perspective to support physicians in writing non-problematic medical notes for primary and secondary usage.

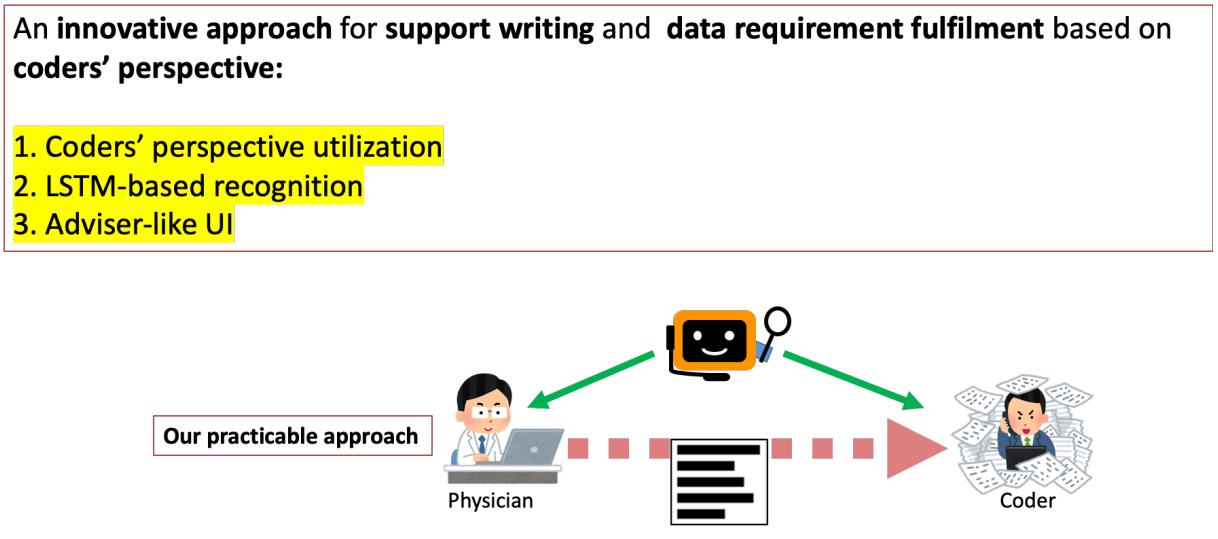

FIGURE 7.1: Our innovative approach

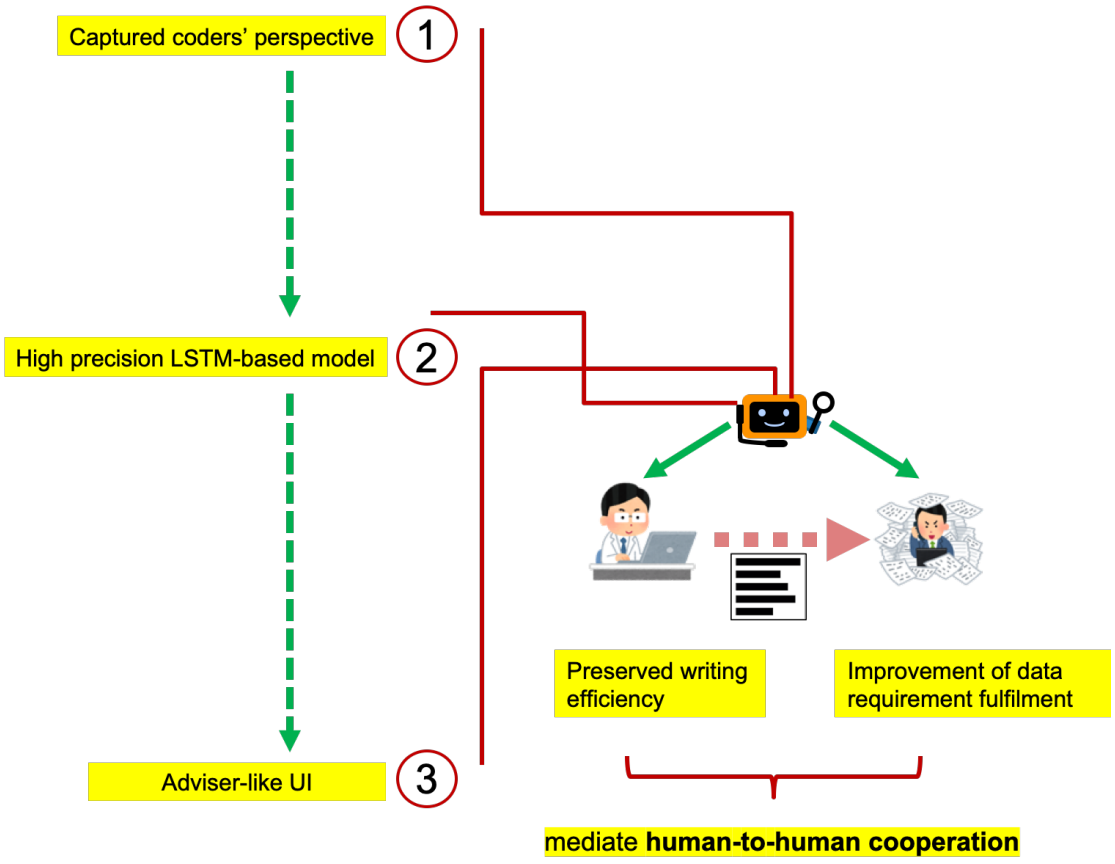

FIGURE 7.2: Our development contributions and its impact

7.1 Summary of findings

In this research, a video-based survey was used to find from the coders' perspective what features or requirements should be included in a system designed to support physicians in writing medical notes. A high precision detection model based on LSTM with an adviser-like user interface (UI) was created. The support system improved physician's writing efficiently without being a burden on the physician. It also increased the amount of structured data wanted by coders, thereby facilitating the medical coding task. Figure 7.2 shows the system's contributions.

There are several writing support systems such as auto-completion and auto-correction or "auto" system currently widely used. However, in the case of medical notes writing, the support is suitable if it is an adviser or reminder system rather than an "auto" system. Reminder systems make enquiries for confirmation or give suggestions, such as "Please add the temperature data" or "Please change df to dengue fever". They allow the physicians complete freedom in expressing their words, while at the same time encouraging them to make changes that suit coders. The approach focuses the physician's mind [132], thereby improving their writing [133].

In case of informal writing styles like those used in Twitter and other Social Networking Services (SNS), writers prefer to take full control of their writing without being bothered by auto-completion or other predictive text support [134]. Previous studies have shown that many writers disable predictive text function in smartphone's keyboard because they find it bothersome [135]. Medical note writing is similar in that text automation that hinders their own writing style is seen by physicians as bothersome. Although the primary purpose of text automation is to speed up the user's writing and make the written texts more understandable, in practice, it may induce errors in written texts due to inaccurate predictions [136] and sometimes decreases the writing efficiency because the users or writers have to manually correct the incorrect predicted text. Written text automation needs to be 100% accurate to be useful in informal writing context because most users do not proofread their texts. It is different in the case of formal writing styles, such as in emails, where smart composition is useful [137].

7.2 Correlation between coders and system's features

The coders' experience influences the specification of the system's features. Coders with more expert education such as coders in the secondary hospital as shown in Table 4.1 are able to assess problematic SOAP notes as being non-problematic, one as shown in Fig. 4.4. In that case, the writing support system used by physicians might be useful for the novice but not the expert coder. However, expert coders may influence the system by providing common helpful keywords and abbreviations that improve writing support system as a guide to physician's note writing.

From the coders' perspective, not all texts written by physicians are problematic, even if they are in the same category as abbreviations. For example, ROM is a general and standardized abbreviation understandable to both physicians and coders. Therefore, it is not problematic and should not be flagged up to the physician. This is because unnecessary notifications might burden physicians while writing medical notes [138]. From this point of view, the more experienced the coders, the less an agent should present notifications to the physicians. This could result in a better writing support. Hospitals could periodically update the list of abbreviations that are acceptable for use in SOAP notes based on the experience of its coders. In this manner, the physicians implicitly request coders to enhance their coding skills for better physicians-coders collaboration.

By investigating the coders' perspective, it was found that the features of the writing support system are not always "assertive" as are auto-completion and auto-correction, but also "encouraging" like the helpful keywords detection discussed in Section 4.3.

Multiple problems in SOAP notes, as shown in Figs. 4.5 and 4.8 require multiple features in the writing support system. A multiple feature writing support is one that combines several features into an integrated system that can solve multiple problems simultaneously. When there are more actions by the system, there is a heavier cognitive load on and greater disturbance to the users or physicians [139].

In order to support the physician in multiple ways when they write, the SOAP note Creation Panel and the Suggestion Panel were separated (Fig. 6.2). This design was adopted in recognition of cognitive load theory, simultaneous presentation of information can cause a redundancy effect [140], whereby the user's action is adversely affected when compared to the sequential presentation of information.

7.3 Positive effects of the writing support system

As presented in Chapter 6, the evaluation of the system showed its positive influence on the physicians' writing efficiency and the amount of relevant data collected by coders. The time necessary to revise problematic notes was shorter with the support system, and the corrected notes had more useful information, according to the coders' evaluations. There was a tendency for physicians to write faster and input more data required by coders. This shows that the system allows physicians and coders to share a common perspective in dealing with problematic notes. The support features allowed physicians to improve the quality of their notes, from the coders' point of view, while not suffering an additional burden.

It is important to improve clinical documentation such as in a Clinical Documentation Improvement (CDI) program, which aims to incorporate the terminology needed to accurately translate a patient's condition into precise codes [141]. One purpose of medical record documentation is to determine the amount of reimbursement for physicians and hospitals. After the documentation is translated into the alpha-numeric codes submitted in claims, the data are analyzed to generate results on quality and clinical outcome measures, in addition to payments. For example, codes reflecting severity of illness and risk of mortality are used to risk-adjust data in order to avoid penalizing physicians who care for sicker patients. This levels the playing field for all physicians. Therefore, physicians' documentation, such as SOAP notes, must be translated into precise codes that will fully reflect the illness severity and mortality risk, intensity of services provided, and resources expended in caring for patients [142]. However, since coding rules and terminology may differ from common clinical language, there is a risk that clinical reality will get lost in translation. This is where CDI programs come in [143].

A CDI program is a comprehensive, multi-disciplinary, hospital-wide effort to incorporate the terminology needed to accurately translate a patient's condition into precise codes [144]. The key players on a CDI team are clinicians and coders. Other members include those in nutrition service, wound care, care management, and laboratory staff. Physicians in this case include any health care professionals licensed and credentialed to diagnose and treat patients. Clinicians play the most important part in CDI since they are familiar with the patient and the conditions being treated. Their documentation drives and controls everything that happens subsequently [145]. The coders' role is, through a medical record review, to capture pertinent physician documentation while the patient is in hospital. The next step, if needed, is to submit a request (query) to physicians for clarification or additional documentation that would permit assignment of a more precise code. Coders should conduct verbal discussions with physicians whenever possible for more effective communication.

Collaboration and exchange of information between a physician and a coder are necessary to ensure that the physician's documentation is actually translated into the codes that reflect the patient's condition. The coder also facilitates clinician education by giving brief presentations at medical staff meetings and conferences as well as by having direct conversations with physicians. Coders must not only collaborate with physicians but must be trained in clinical terminology and diagnostic criteria most often encountered in the CDI process. There are almost always opportunities to improve code selection, sequencing, and application of coding guidelines.

From the CDI perspective, physicians and coders should work together intensively. Physicians can learn standardized documentation from coders, while coders should also improve their expertise in the coding task. A technical approach, such as documentation or a writing support system for Clinical Physician Order Entry (CPOE) could accompany the CDI program by virtually bridging physician-coder collaboration to produce improved clinical documentation for optimization of the coding task. This bridging should naturally mimic what happens in the physician-coder relationship. In real practice, the coders have a chance to teach physicians about making standardized documentation for the coding task based on the coders' perspective. Based on real practice, the writing support should start from the coders' perspective as discussed in Chapter 4. The physicians have an option to learn from the coders, as reflected by the system being adviser-like, as discussed in Chapter 6. The physicians also have the option to write medical notes that are more standardized, based on the coders' perspective.

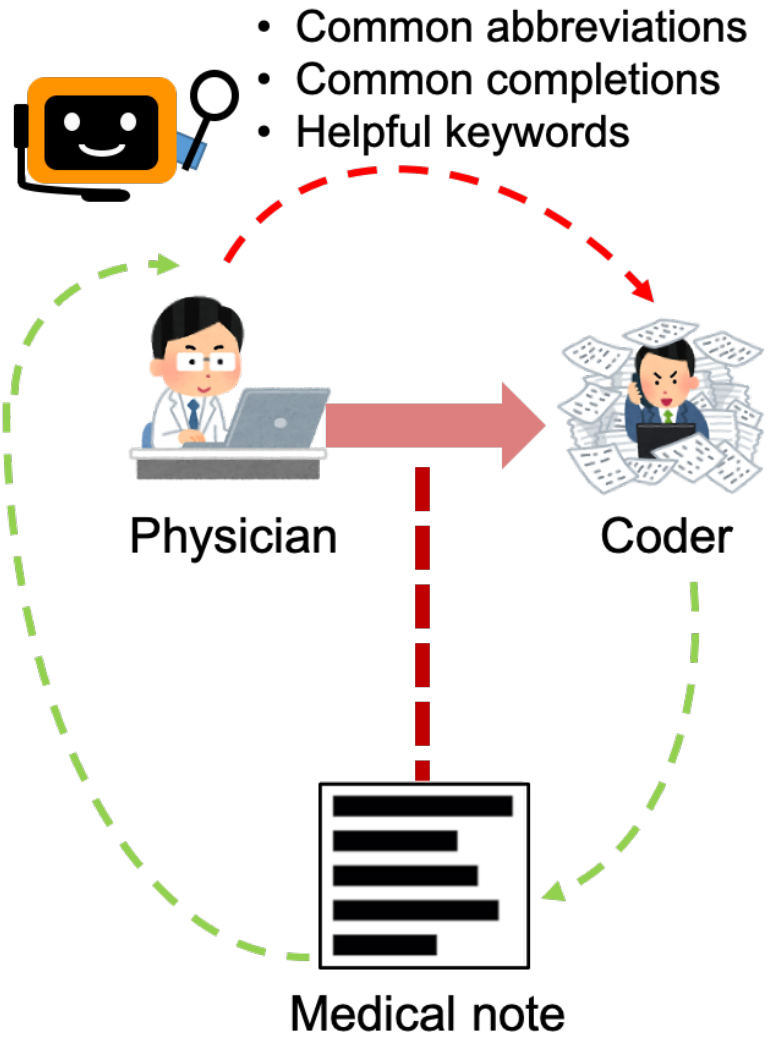

FIGURE 7.3: Interactive cooperation between physicians and coders through the agent system

7.4 Interactive cooperation between physicians and coders through an agent

Physicians and coders tend to cooperate by mediating through an agent. Although in real situations they tend to separate their roles, with the agent their knowledge seems combined and can teach each other and form a consensus that is understandable by both parties, such as common abbreviations and helpful keywords, as discussed in Chapter 4. Figure 7.3 shows the interactive cooperation between physicians and coders through the agent system.

The initial interaction from our approach is one-way interaction, in which first the physicians write medical notes, then coders advise through the support system's suggestion list of what should be improved. However, based on the coders' perspective, such as

helpful keywords and smooth utilization of the support system by physicians in adding completions and correcting abbreviations under its guidance, the one-way interaction becomes two-way interaction. In the two-way interaction, the physicians not only follow exactly what the system suggests, as with a template approach [39], but physicians also write additional information based on their own perspective, which may be useful for learning how to further improve the system. With the system's learning improvement, the writing support system could suggest more appropriate support to the physicians, such as more detailed completion and filter out rare abbreviations, while also preserving the coders' requirements of structured data. It is believed that the freedom to write while guided by the system is the reason for the positive effect on the physicians when writing medical notes with the support system.

In some hospitals, experienced coders might have more say in the coding process than do physicians, and quasi or part-time coders might have insufficient knowledge to deal with complex or unusual coding cases. Experienced and less experienced coders may interact differently with physicians. For example, an experienced coder might raise critical enquiries with a physician and input what they deem to be the appropriate code based on supporting evidence in the medical records. In contrast, inexperienced or part-time coders may have more shallow enquiries due to lack of knowledge resulting in the entry of codes that are less accurate.

Such interaction may occur due to what is well-known as a power or authority gradient, which describes the established or perceived chain of command and decision-making power or hierarchy in an organization, and how this affects members according to their experience. If the power is concentrated in one person there will be a steep gradient, whereas a shallow gradient is typical of more democratic and inclusive involvement of all members. In physician and coder interactions, power may be overly concentrated in the physician. To make this interaction more democratic, a computer-mediated support system representing the agent that incorporates the coders' perspective would achieve a cooperative shallow gradient of interaction. With this computer-mediated support, the interaction between physicians and coders are equally preserved, which is proposed as the 'ideal' interaction among them.

7.5 Trade-off between existing approaches and the proposed approach

Transforming medical notes into structured data for primary and secondary usage of clinical data is not a trivial process. Using a template is the common approach, however

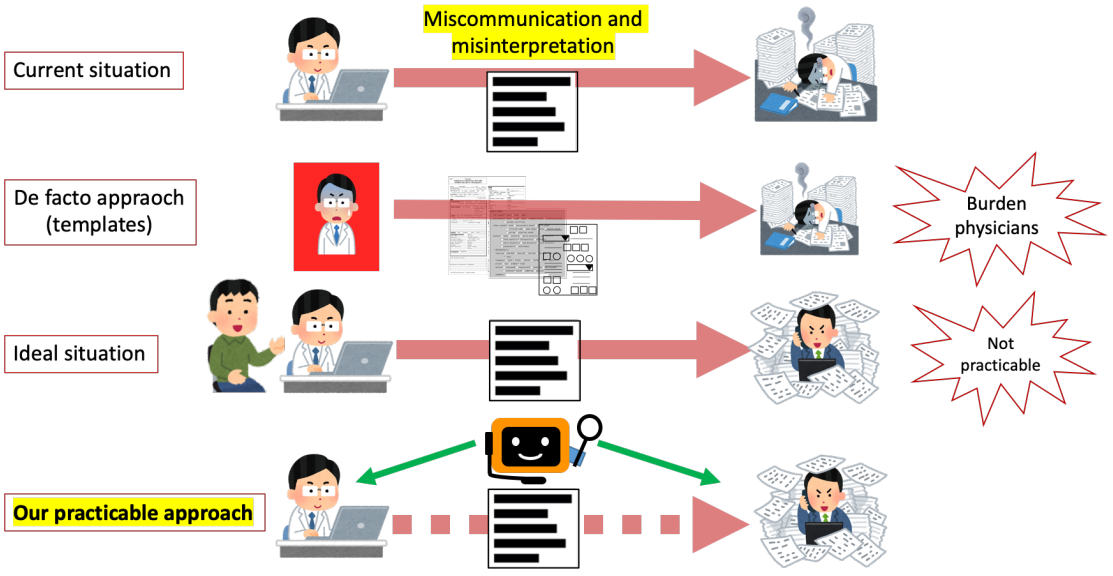

FIGURE 7.4: Trade off of our approach

it may not seem useful to the physician. Our approach eliminates this issue, while obtaining structured data from the physician's medical notes. Figure 7.4 compares our approach with other existing approaches.

In our understanding of the coders' perspective, the writing support system follows the natural interaction between physicians and coders doing the medical coding task. The natural interaction is able to support both parties: the physician's writing efficiency is increased and more of the structured data required by coders is obtained.

The natural interaction seems effortless according to the physicians, as shown in Table 6.2 and Appendix E. Effortless is related to effort expectancy that had been introduced in the Unified Theory of Acceptance and Use of Technology (UTAUT) model [16], and is a crucial predictor of technology acceptance. According to this study [131], effort expectancy is "the degree of ease associated with the use of the system". UTAUT model is the most effective model for analyzing technology acceptance [146]. The UTAUT model consists of six main constructs, namely performance expectancy (PE), effort expectancy (EE), social influence (SI), facilitating conditions (FC), behavioral intention (BI) to use the system, and usage behavior, as shown in Fig. 7.5. The UTAUT model contains four essential determining components and four moderators. According to the model, the four determining components of BI and usage behavior are PE, EE, SI, and FC. Gender, age, experience and willingness to use are moderators that affect usage of technology, as shown in Fig. 7.6.

From the UTAUT model, effortless usage of a new system such as our writing support system can further positively influence the physician's behavioral intention in writing

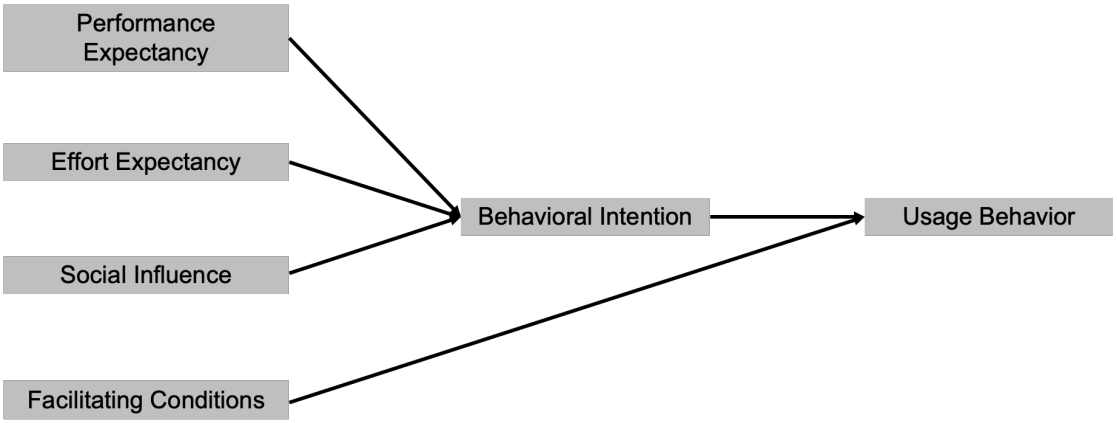

FIGURE 7.5: UTAUT model consists of six main constructs
[131]

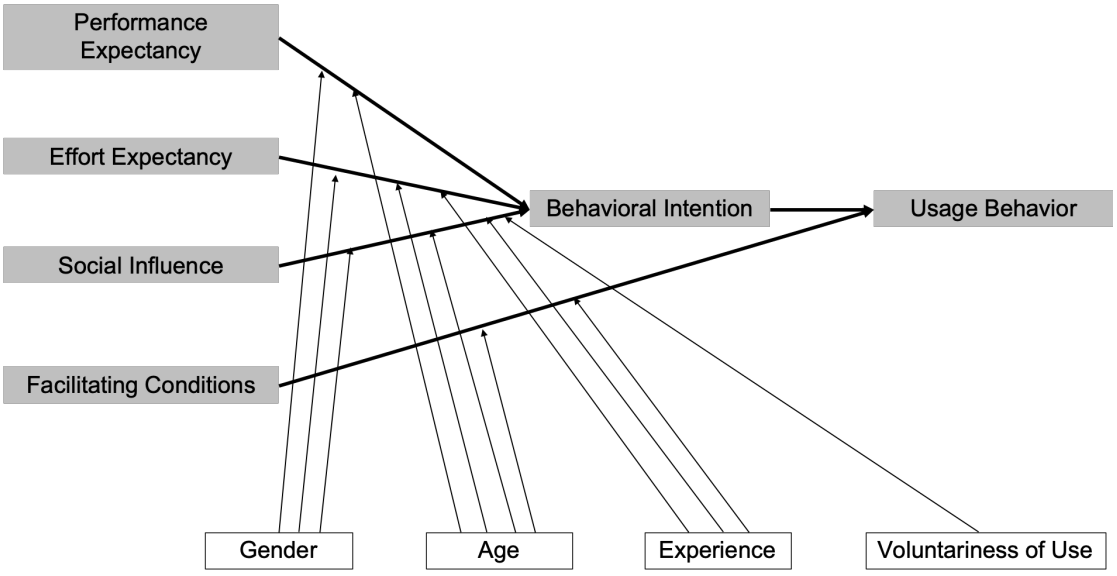

FIGURE 7.6: The UTAUT model contains four essential determining components and four moderators
[16]

medical notes. It means that new processes smoothly align with existing work processes. This is because one of the often-cited barriers to adoption of a new system is the misalignment of new processes, like our support system, with existing work processes, such as data entry or writing medical notes. The physician's effortlessness shows that the new processes align with their existing work processes. The majority of the studies indicated that physicians are more concerned with integrating a new process into their existing clinical processes than on the need to fundamentally alter those processes [147]. Positive behavioral intention triggers positive usage behavior by physicians that fulfil our writing support system's goals.

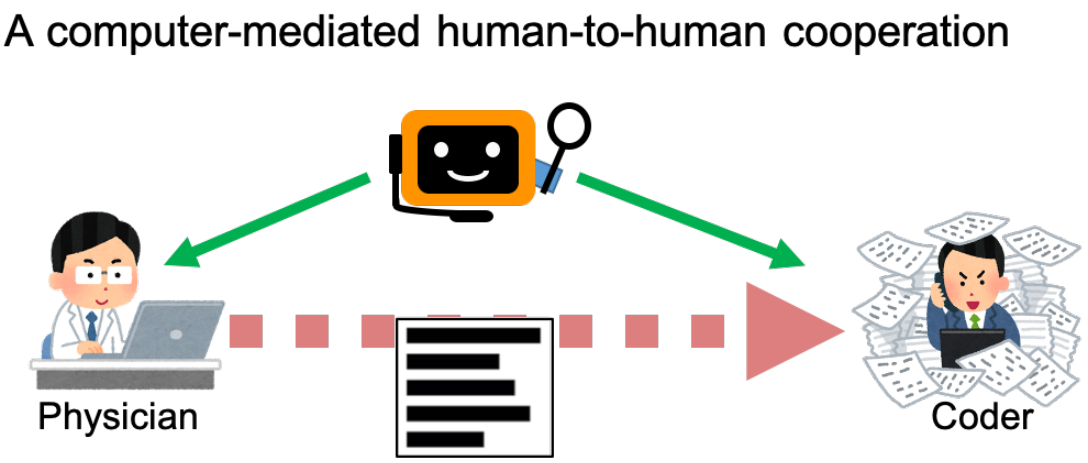

FIGURE 7.7: Computer-mediated human to human cooperation

Based on our research process, it was shown that our approach is innovative in transforming human-origin notes to machine-readable data via computer-mediated human-to-human cooperation. Figure 7.7 shows our innovative approach to transforming medical notes into structured data for primary and secondary usage of clinical data.

7.6 Generality

This study can be applied to other domains, which require collection of structured data from free text writing, such as automation of general administrative documentation and bureaucratic work, which until recently has been a burden on human-to-human and human- to-machine cooperation.

The configuration of the physician and coder cooperation is often found in other fields, such as between academic authors and reviewers. The author is the physician and the reviewer is the coder. From that point of view, the computer-mediated support can support the author in writing a paper based on the reviewer's perspective. Thus, it can be generalized as a human-to-human interaction-oriented support for documentation based on data from the requester's perspective.

According to the authority gradient, the human-to-human cooperation may not be ideal due to authority differences among team members. In cases where all team members should be treated equally, such as in the case of co-writing documentation, there should be a system to preserve the team members' equality. It is important to ensure the optimal outcome of human-to-human cooperation, such as a document that fulfils all team members' requirements and intentions.

7.7 Limitations

The dataset for the research used pseudo and free public medical notes, due to issues of privacy and confidentiality.

Interactions including nurses/physicians or other clinical teamwork settings and data input devices such as large screen or a tablet were not examined for comparison.

The study only involved coders in Indonesia. Research among physicians and coders in several countries is advisable.

7.8 Impact

The first major impact of our research was to reduce the dependency on templates which may burden physicians. Additionally, with the possibility of unlimited replication of coders' representatives, the proposed system enables support of both physicians and coders. It can also increase acceptance of the EMR entry system by physicians, and improve physician-coder interactive cooperation, which is critical for transformation of human-origin data into machine-readable data.

This work will also support exploration of new medical note entry strategies, such as entry with natural human-to-human interaction, and accommodate multidisciplinary perspectives based on the above findings, and will even encourage the development of more usable applications for establishing and sustaining better transformation of human-origin data to machine-readable data by consideration of those strategies. It also fosters social cooperation among healthcare professionals, which is necessary for smooth patient care and hospital operation [17].

Effective teams are characterized by trust, respect, and collaboration. Teamwork is essential in a system in which all employees are working for the good of a goal, who have a common aim, and who work together to achieve that aim [148]. When considering a teamwork model in healthcare, an interdisciplinary approach should be applied in which most teamwork hurdles can be overcome with an open attitude and feelings of mutual respect and trust. A previous study has determined that improved teamwork and communication are said by healthcare workers to be among the most important factors in improving clinical effectiveness and job satisfaction [149]. Healthcare teams that lack trust, respect and collaboration are more likely to make mistakes that could negatively impact on the safety of patients and thwart hospital management goals. Trust is an important factor in bridging social cooperation among healthcare professionals. Maintaining a high level of trust among healthcare workers, such as between physicians and

coders, is crucial. A solution must be found when a high level of trust is difficult to achieve due to lack of communication and misinterpretation of other people's perspectives. One way of solving this issue comes from information technology: the computer can be used as a mediating tool to bridge cooperation between healthcare workers virtually. A previous study [150] has shown that the more the users trust an agent or computer-based representative, the more likely they want to consciously impress the agent. This finding seems consistent with the theory that people apply the same social rules to computers as they do to humans [151]. In other words, people often want to impress someone whom they trust (*e.g.*, parents and spouses). This trust effect also occurs when physicians write medical notes using a support system as discussed in 6. With the suggestions provided by the support system, the physicians' impression was manifested by the improved utilization of the support system to make non-problematic medical notes, as shown in Table 6.3.

In future research, as the writing support system based on the coders' perspective is still in its infancy, there is the possibility of expanding it to accommodate the perspectives of multiple healthcare workers, such as pharmacists and nurses, and integrating them to establish a wider range of natural entry support, such as using speech and gesture. In this future scenario, the writing, speech and gestures from the physicians could be interpreted by the support system to produce structured and standardized data required by multiple parties. The support system could also suggest required data to the physicians by using text or speech or the system's gesture, which may be in the form of a robot or virtual agent. The support system itself can be equipped with an artificial intelligence (AI) avatar [152] to make it a more human-like supporter.

The support system should become the physician's partner. Ideally, physicians should spend minimal time on bureaucracy, as it reduces their time for primary tasks, which is treating the patient. Although documentation is useful for tracking and logging medical records, it should only be a minimum part of the physician's routine tasks. If other parties require structured data from medical notes, it should be done via minimum invasive methods with a precise requester perspective, such as computer-mediated support for writing medical notes from the coders' perspective. Without a precise requester perspective and natural flowing entry mechanism, the physicians may still be burdened and the required data not provided.

Chapter 8

Conclusion

A common way to optimize the medical coding task is by developing a structured data entry system or template-based SOAP notes [39]. However, templates often require many fields and take too much time for physicians to complete the form. In addition, physicians are more familiar with free text-based notes, as explained in a previous study [5]. Another approach is co-writing of SOAP notes by the physician and coder. However, that approach seems impractical due to the lack of coders preventing a one-to-one co-writing partnership.

One possible approach is to retain the freedom of physicians to write free text-based SOAP notes while developing a support system to enable the SOAP notes to be written in a way that poses no problems to coders. With this objective in mind, a new innovative approach was designed based on the coders' perspective. A support system for physicians writing medical notes without hindering their natural way of writing was made and evaluated.

In this research, first, coders' perceptions of problems with medical notes was collected using a video-based survey. Second, an LSTM- based high precision detection model was constructed. Third, an adviser-like user interface (UI) was designed and confirmed to support physicians and coders. The physicians were able to write medical notes more efficiently and medical coders were able to obtain from the notes more of the data that they required.

The video-based survey revealed that a support system should be advisory rather than assertive, as happens when coders and physicians work together. Each country might have different configurations of medical coding tasks, but the purpose is the same, to optimize the clinical data for primary and secondary usage. A support system is required for the coding task due to the multi-task nature of the physician's job, which makes it

very likely that they could forget to include specific data. The system can remind them to include necessary information as they write the notes. At same time, the coders are required to ask physicians for clarification of any uncertain or lacking information in the notes. The coders have expertise in transforming human-origin data into machine-readable data or transforming unstructured text into structured and standardized data. Coders can better understand the physicians' notes and requirements by interacting with them, and if necessary guide the physician's writing based on the coders' perspectives. Understanding the data requester's (coders') perspective is important in this task because it replicates the natural interaction between the coders and physicians in real life. By taking into account the coders' perspective, the system mimics the interaction that happens in real practice, without negatively impacting on the physician's writing efficiency.

The support system understands the coders' requirements from the physician's medical notes. For that, first it uses a detector or classifier to recognize relevant information that should be extracted. For instance, unknown abbreviations were one of the main concerns of coders. Detecting abbreviations from the medical note is a challenging task because they are often written using an informal style, with noisy entities that are difficult to recognize. Nevertheless, the LSTM-based classifier produced promising results in that task. In addition, the classifier was able to perform well even with class imbalance, which is a common challenge in this kind of text processing task.

After the support system understands the coders' requirement from the texts, it can then be used to support the physicians as they write the medical notes. To interact with the physician, the support system must have a user interface that accommodates the natural way in which the physician writes. When the system detects some missing required data, it must be presented as a suggestion list that the physician may optionally accept to follow. The physicians had freedom to choose whether to use the support system, which is primarily designed to fulfil the coders' requirements. With this freedom to choose, the physicians virtually interact with the support system as natural as when the physicians and coders interact in real practice during the medical coding task. The natural interaction using a computer-mediated support for writing medical notes with the coders' perspective taken into account was advantageous to the physician and coder alike.

8.1 Main contributions

The achievements and processes to create an agent-based system in this study could be applied to other domain tasks that require transformation of human-origin data

into machine-readable data. The first contribution is process of a video-based survey to understand the coders' perspective in dealing with problematic SOAP notes, enabling the design of a medical note writing support system for physicians. The second contribution is high precision deep learning modelling using LSTM to detect *informal* abbreviations from medical notes, which is a vital task for structured entity identification in EMR. Additionally, our approach expands the field of application of the LSTM model since the technique has never been used for this kind of task. The third contribution is the design and evaluation of the novel writing support system's user interface. The evaluation compared it with a conventional medical notes' user interface in term of its efficiency to write medical notes and in terms of its usefulness to capture more of the data required by the coders. In summary, the main contribution is the system's ability to simultaneously capture clinical data for primary and secondary usage without sacrificing the physician's typing manner and efficiency. This doctoral thesis shows that the balance between natural flow of entry and requester's precise perspectives produced the best of both worlds, the physicians' and the coders'. The approach harmonizes two aspirations.

Although the initial focus of this research was to tackle problems encountered in the medical coding task in Indonesia, it pertains to a wider global issue, such as informal abbreviations in medical notes, which are commonly encountered in several countries such as USA, UK, Israel, Thailand, and Japan [153] and the physician's burden in entering data via template-based medical notes [154].

Appendix A

Physicians preference in entry SOAP data

In order to find out more the SOAP note entry problem, a small usability trial in Indonesia as our case study was conducted, using subject doctors from private hospital, public hospital and academic hospital given a task to complete SOAP note for the case of pediatric flu-like symptom in Indonesian language.

In Indonesia, currently exist two type of SOAP note consist of simple SOAP note (SiSOAP) as shown in Fig. A.1 and standard SOAP note (StSOAP) as shown in Fig. A.2. Both are using fully free or narrative text format that is common to use for the entry process. The new introduced format is template-based SOAP (TSOAP) as shown in Fig. A.3. The doctor's task for this small trial is the doctor anamnesis the simulated patient, then inputted the observation to the three types of SOAP note formats, 1) simple SOAP note (narrative format with less detail fields), 2) standard SOAP note (narrative format with more detail fields), and 3) template-based SOAP note (structured data entry format). The trial result is shown in Fig. A.4.

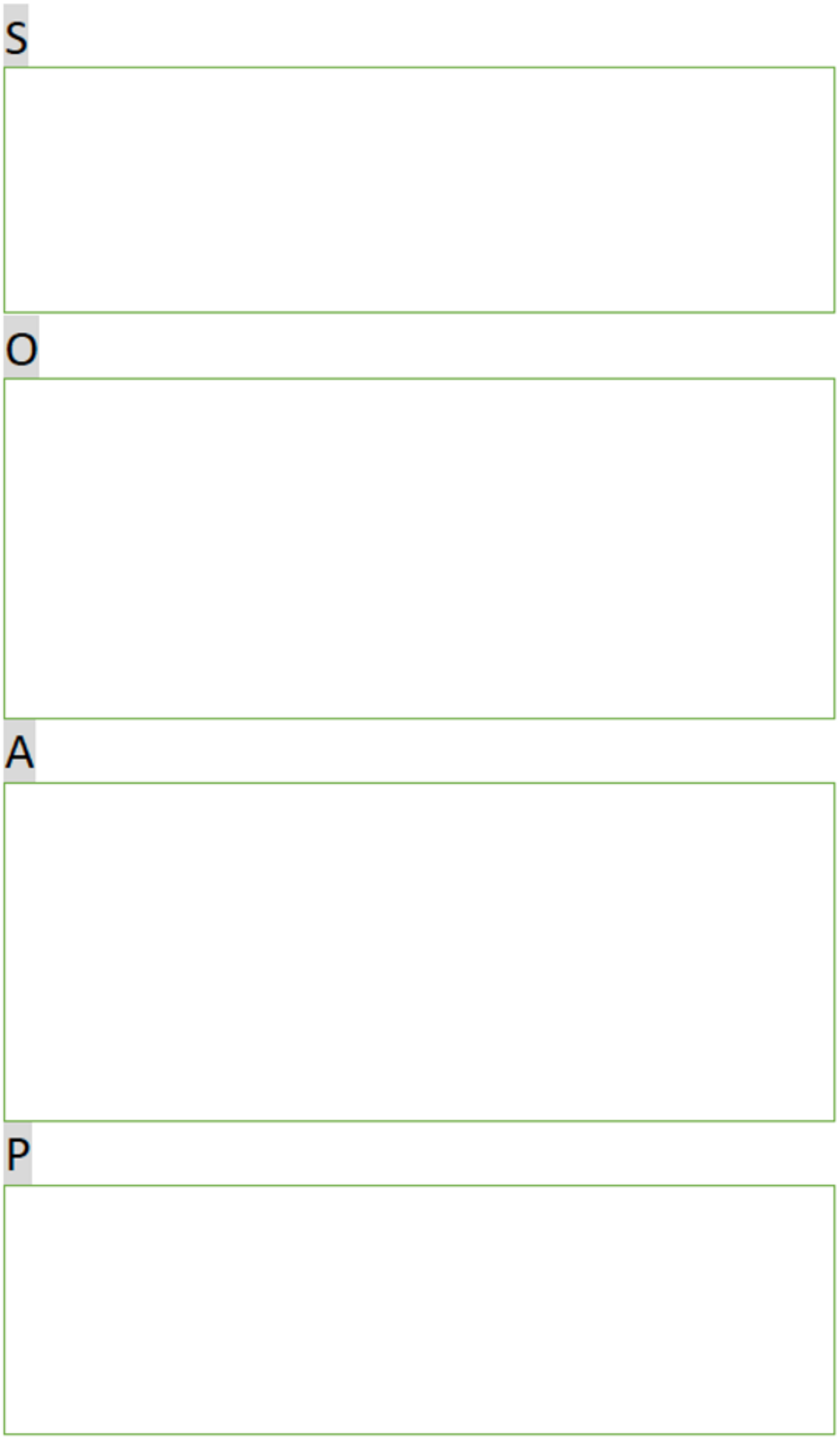

FIGURE A.1: Simple SOAP note format

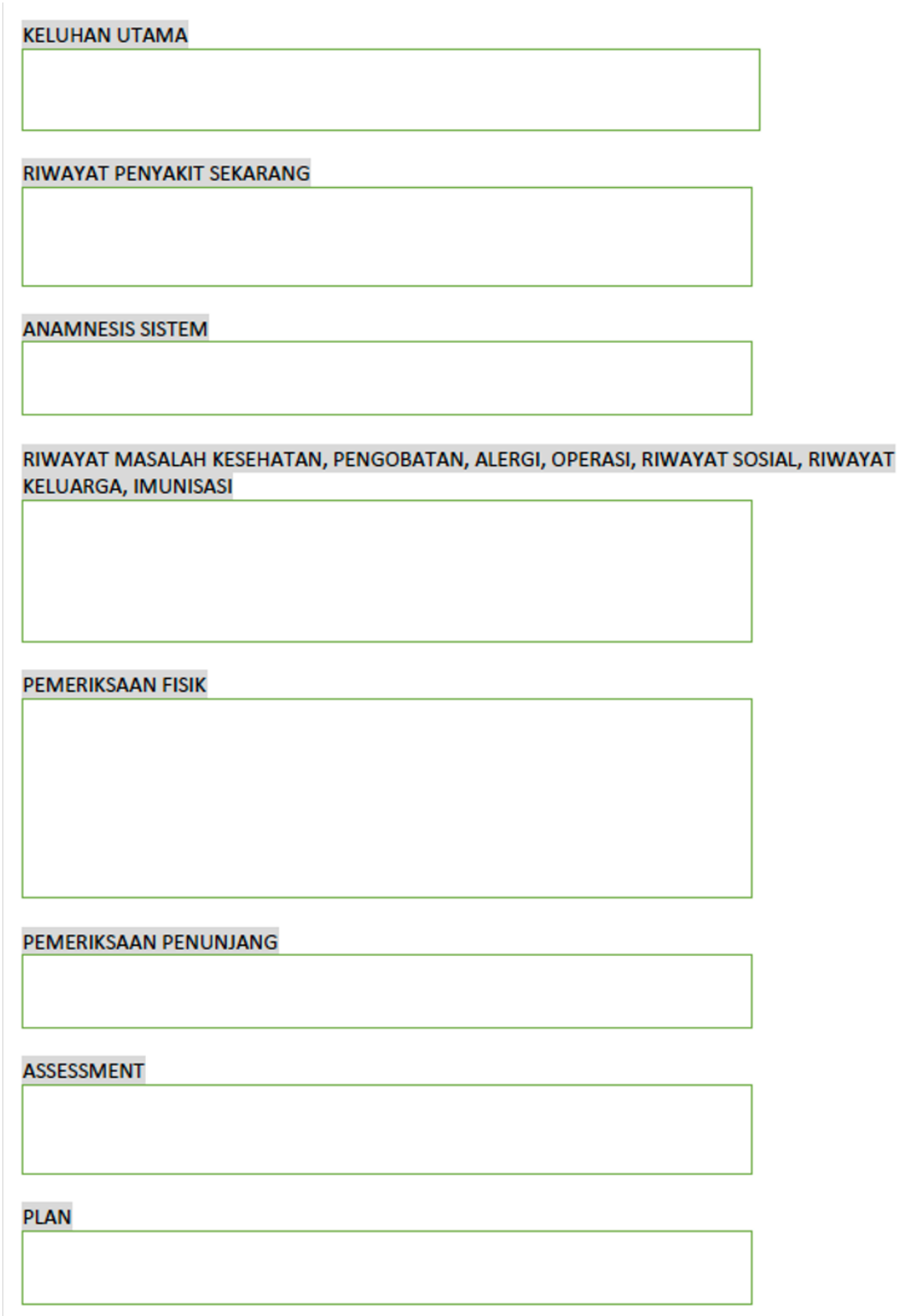

FIGURE A.2: Standard SOAP note format

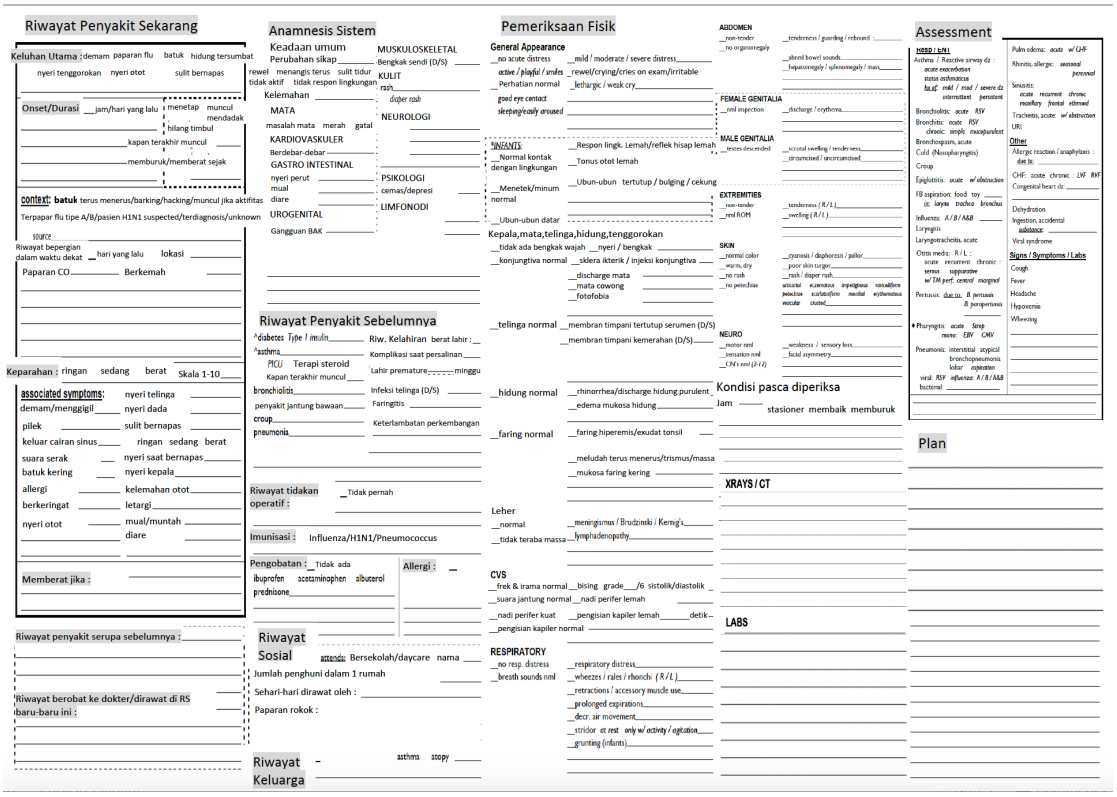

FIGURE A.3: Template-based SOAP format

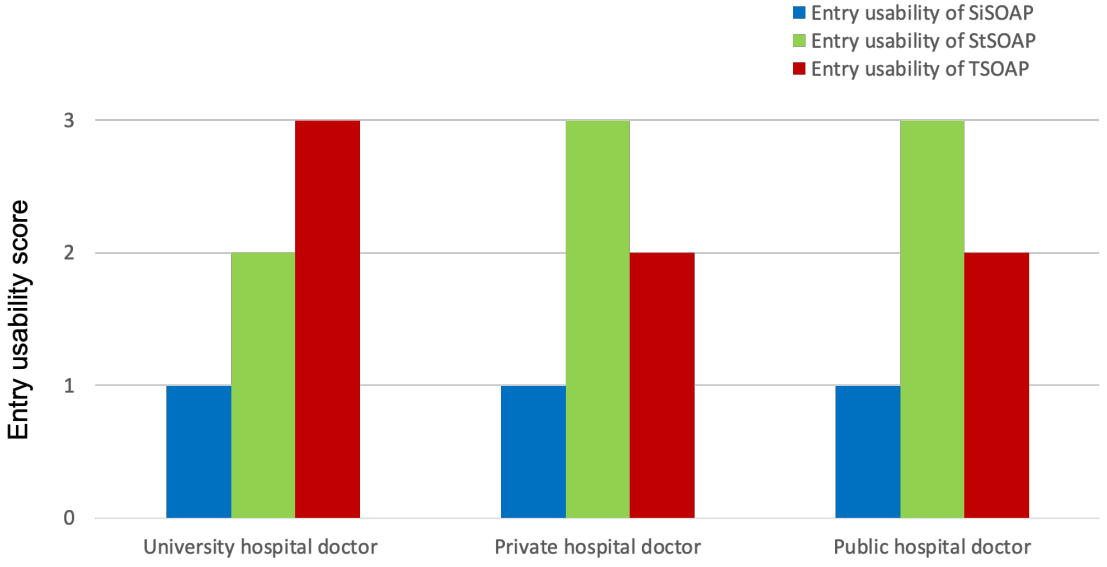

FIGURE A.4: Entry usability scores of three types of SOAP notes

Appendix B

Indonesian pseudo SOAP notes

In the video-based survey to understand coders' perspective in dealing with the problematic SOAP notes, an 11 pseudo SOAP notes in Indonesian language for the experiment was provided. The 11 pseudo SOAP notes follow.

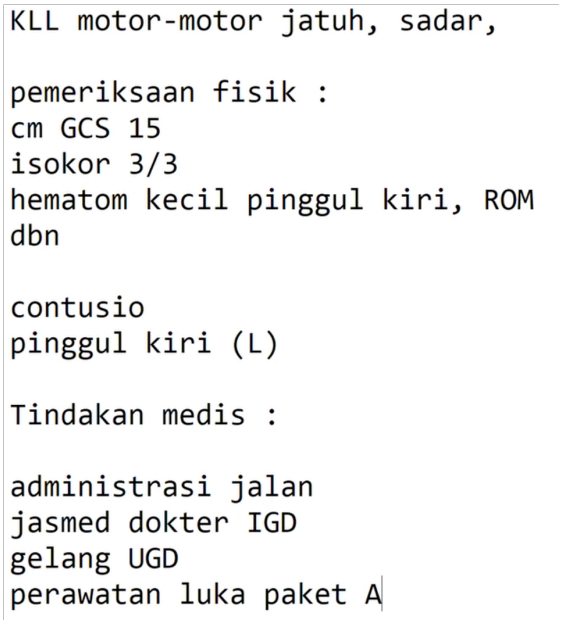

Figure B.1: SOAP 1

```
dikatakan setelah minum gabapentin kedua tangan bengkak, nyeriseluruh badan,

TD 90/50
Thorax : SDV +/+N, Rh -/-, Wh -/-
Abd : BU+N,NT-

Neuralgia post herpetik (L)

Gabapentin dihentikan dl

PULANG :

NEURODEX NO X
S2DD TAB 1
-
ANALSIK NO X
S3DD TAB 1 PC
```

FIGURE B.2: SOAP 2

```
2 hari badan menggigil, pusing batuk
sudah minum panadol

s/38,7
t. 140/87 mmhg

SUSP VIRAL INF H2 DD; DF, TFA (B)

EDUKASI
MEDIKAMENTOSA

PULANG :

SANMOL 500 MG TAB.
NO. VIII S TAB 1 SETIAP 6-8 JAM PRN
MUCOHEXIN 8 MG TAB.
NO. VIII S TAB I SETIAP 8 JAM
DEXTAMIN TAB.
NO. X S TAB 1 SETIAP 8 JAM
```

FIGURE B.3: SOAP 3

```
pasien mengatakan kenceng kenceng sejak jam 04 00, belum ada lendir darah
rencana SC, indikasi re SC

IGD. TRIASE KUNING
PSIKOLOGIS CEMAS
RISIKO JATUH TIDAK BERISIKO
RESPON TIME <5 MENIT
STATUS GIZI CUKUP
T 114/77
N 87

MULTIPARA G3 P2 ABO AH2 RE SC DR VONNE (B)

EDUKASI
MEDIKAMENTOSA
PRO SC DR VONNE

Periksa Lab
OPNAME
```

FIGURE B.4: SOAP 4

```
TIBA2 SESEK DALAM PENGOBATAN JANTUNG

ASTMA ATTACK PADA RIW STEMI DLM TX (B)

EDUKASI
MEDIKAMENTOSA
NEBULIZER VELUTIN P;US/FLIXOTIDE 1/1

PULANG :

SALBUTAMOL 4 MG TAB. GEN+
NO. X S TAB 1 SETIAP 8 JAM
METHYLPREDNISOLONE 4 MG TAB
NO. X S TAB 1 SETIAP 8 JAM
CETIRIZINE 10 MG TAB
NO. X S TAB 0-0-1
```

FIGURE B.5: SOAP 5

```
naik motor jatuh sendiri kemarin jam 20.00, sudah diberi betadin di
rumah
ada luka lecet di kaki kanan kiri, tangan kiri, bawah hidung
tidak pusing
menurut ibu, imunisasi disekolah saat SD sudah lengkap

IGD. Triase hijau
Psikologis cemas
Risiko jatuh tidak berisiko
skala nyeri 4
Respon time < 5 menit
status gizi cukup
T 100/60

MULTIPLE VE TIPIS (B)

TL, EDUKASI

PULANG
```

Figure B.6: SOAP 6

```
pipi kanan masih kedutan

psikologis tenang
status gizi baik
tidak berisiko jatuh
t 150/100

TIC FACIALIS KANAN (B)

edukasi

medikamentosa
kontrol 2 minggu
PULANG :

ADMINISTRASI
IMBALAN JASA MEDIS SP

HALOPERIDOL 0.5 MG TAB
3x1 NO 21
-
```

Figure B.7: SOAP 7

```
kontrol rutin TB bulan I 2 minggu pertama pegobatan sudah hampir habis,
4 tab sekali minum
membaik
makan banyak

CM, anak berjalan sendiri baik
pulmo rbh
kel limfe supraclav dan servikal > N

KTPB kontrol (L)

Edukasi
medikamentosa

resep
PULANG :

ADMINISTRASI
IMBALAN JASA MEDIS SP
```

FIGURE B.8: SOAP 8

```
KEL -

TIO 7/5.5

ODS POAG+DRY EYE (L)

RUJUK PERIMETRI

REFRAKSI
FUNDUSCOPY
TONOMETRI
DIRUJUK : RS YAP

GLAOPEN 1x1 ODS
```

FIGURE B.9: SOAP 9

```
kontrol

tf 4 manus s (B)
oaknee d (L)
tfcc lesion d dan oa knee d (L)

inj

inj

FLAMAR ZALF NO 1 2XUE
GLUKOSAMIN TAB NO10 2x 1
```

FIGURE B.10: SOAP 10

```
hidung pilek,sering mengorok,telinga sudah tidak ada keluhan

AS CAE cerumen, mt intak
cavum nasi D/S tampak pembesaran adenoid

AS OMA st resolsi + cerumen (B)
Adenoiditis (B)

evakausi
PULANG :

ADMINISTRASI
IMBALAN JASA MEDIS SP
EVACUASI SERUMEN 2 TELINGA

CETIRIZINE SYR GEN JKN
```

FIGURE B.11: SOAP 11

Appendix C

Informal abbreviation detection model

In the informal abbreviation detection modelling, a seven inputs variation and combination to feed into our bidirectional LSTM model was used. The seven inputs variation and combination follow.

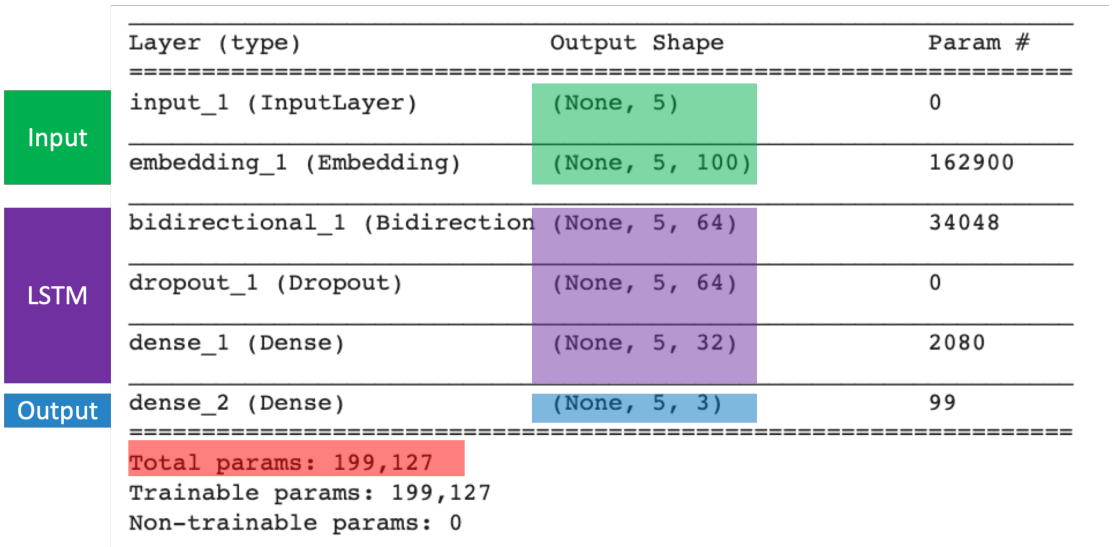

FIGURE C.1: Model with embedding (baseline model)

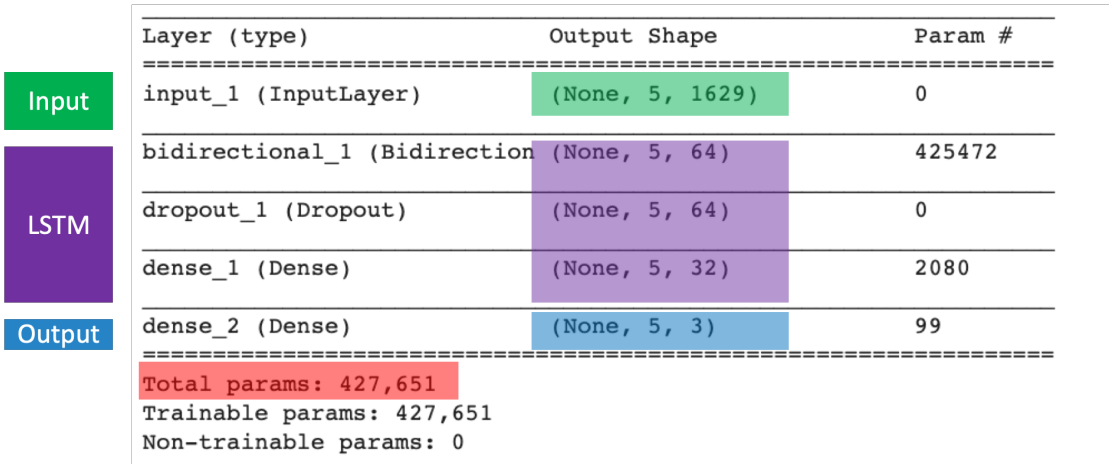

FIGURE C.2: Model with Bag of words (BoW)

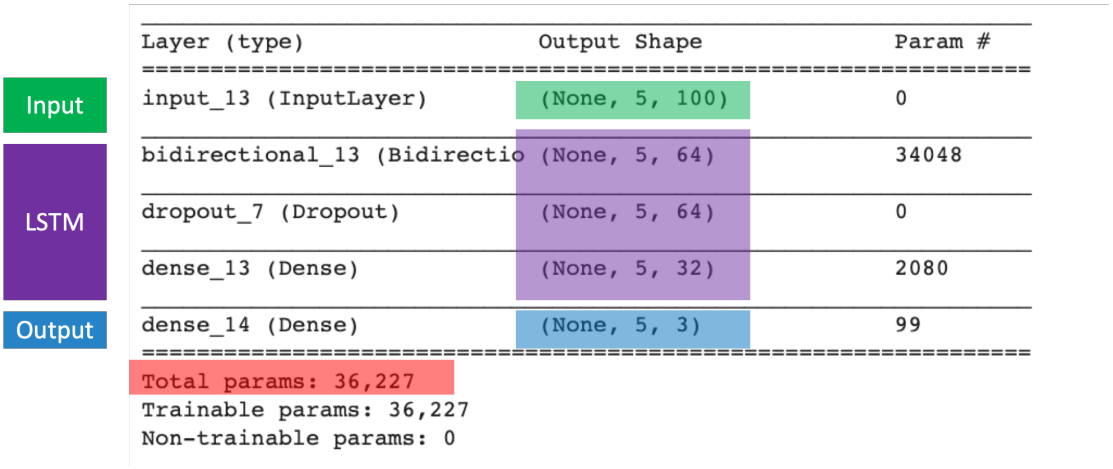

FIGURE C.3: Model with word2vec

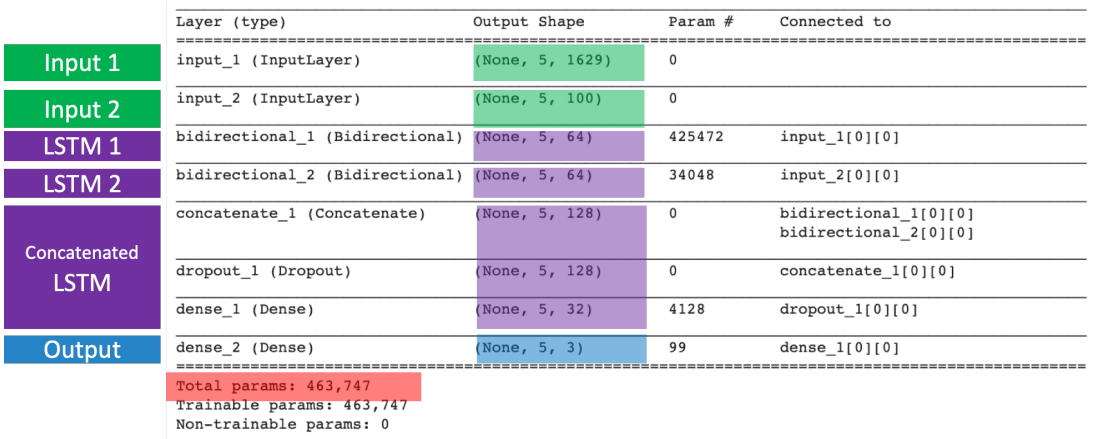

FIGURE C.4: Model with BoW and word2vec

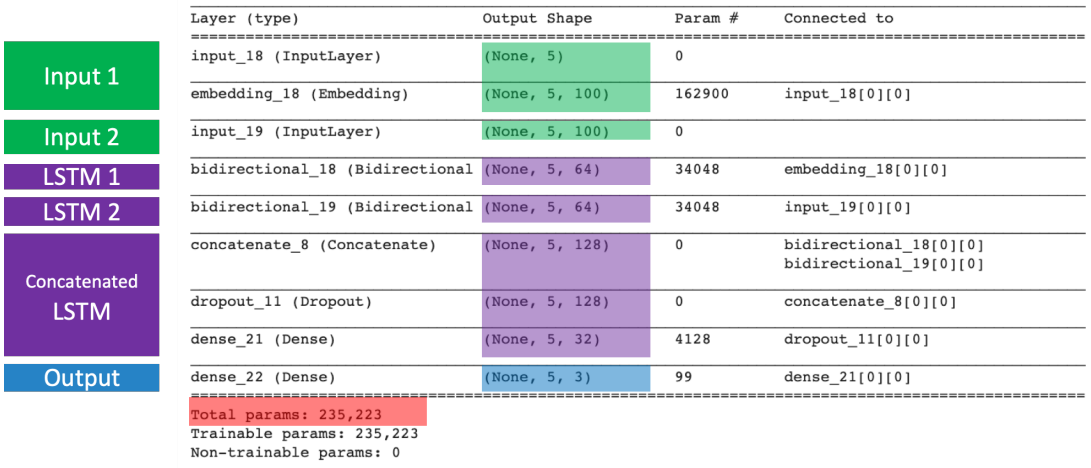

FIGURE C.5: Model with embedding and word2vec

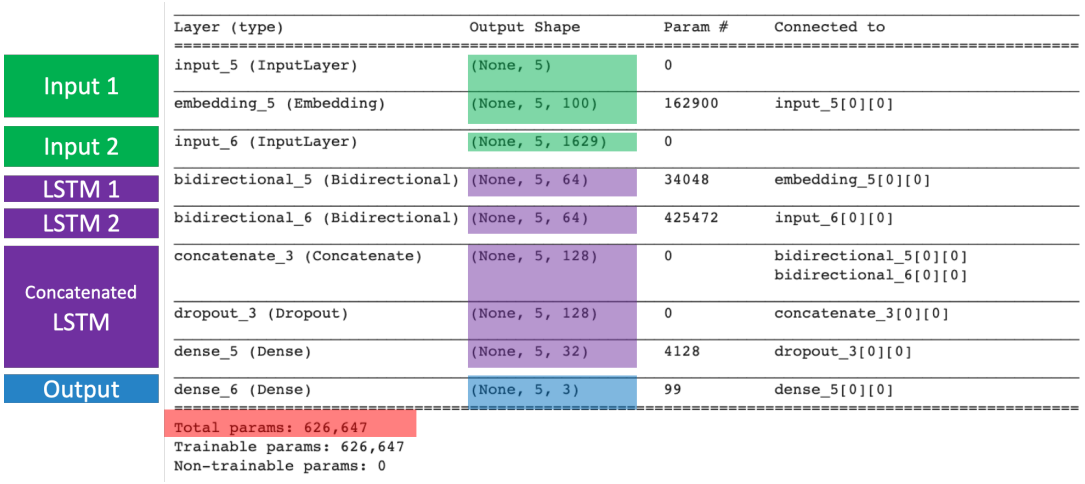

FIGURE C.6: Model with embedding and BoW

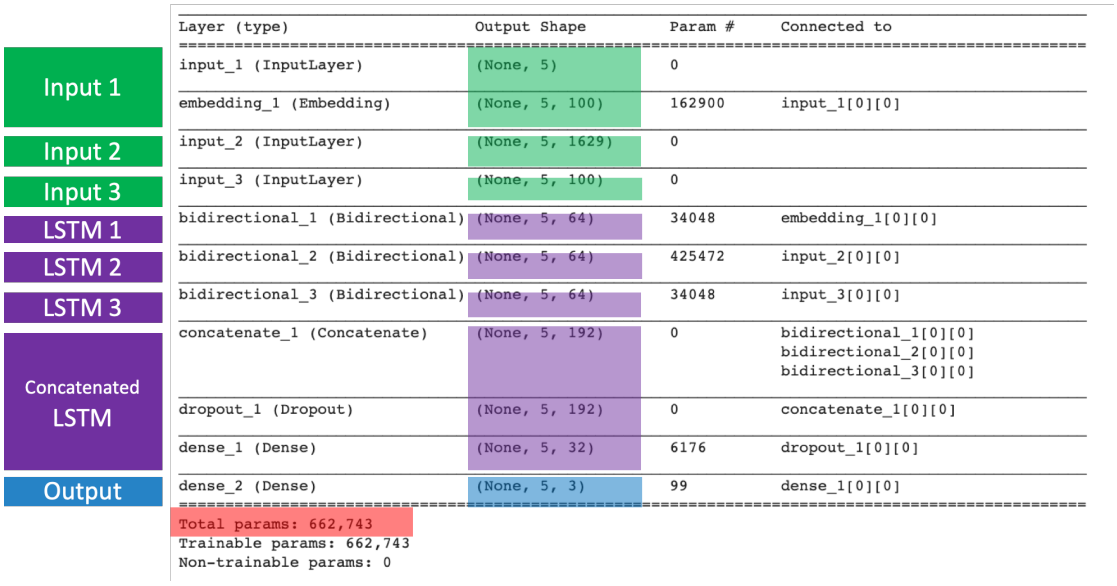

FIGURE C.7: Model with embedding and BoW and word2vec

Appendix D

Writing support system's UI prototype

In evaluating significant differences between SOAP note's writing without support system and with support system, a 20 web pages that contained predefined SOAP notes was used, which divided into two group of prototypes. The predefined SOAP notes are pseudo SOAP notes (Appendix B) that are used in the video-based survey of this research project.

First prototype's group is 10 web pages that are SOAP note editor without support system, and the second prototype's group is 10 web pages that are SOAP note editor with support system. The two group of prototypes is in Indonesian language follow.

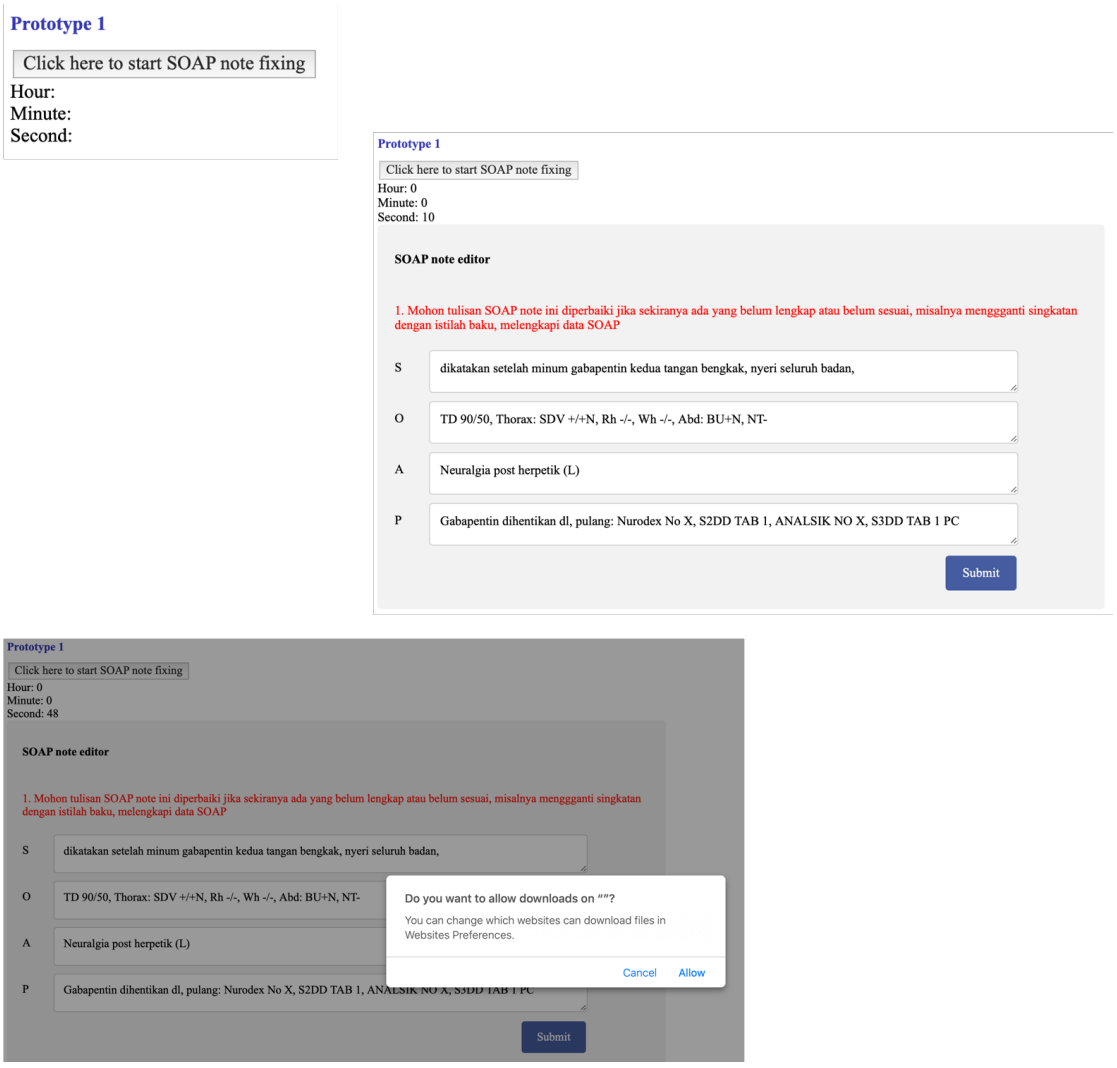

FIGURE D.1: Prototype 1 - SOAP note editor without support system)

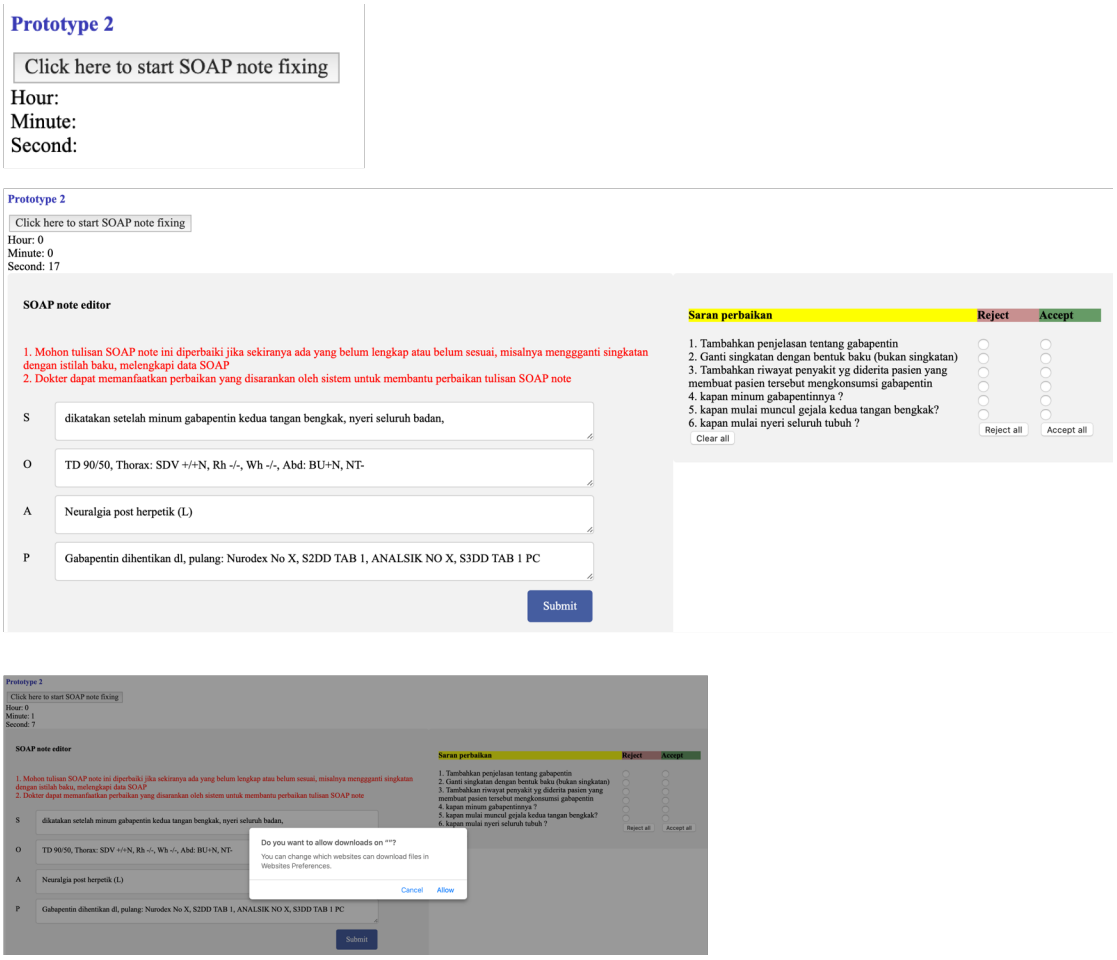

FIGURE D.2: Prototype 2 - SOAP note editor with support system)

Appendix E

Feedback comments about support system

After finished the experiment with our prototypes as discussed in Chapter 6, two Indonesian physicians are asked to give their free comments about our novel writing support system. The summary of free comments (translated) follow.

- Physicians are more comfortable with suggestion list or support system.
- Physicians are more focused with support system.
- Support system reminds some important things that were missed.
- Support system helps increase required attentiveness.
- The more complete or appropriate the suggestion list, the more attentiveness increases to complete and correct the presented SOAP notes.
- Not all suggestions in suggestion list are used, some are incorrect according to physicians, but almost all suggestions are used.
- Although some suggestions are not detailed, physicians are reminded and add details in SOAP note editor based on the undetailed presented suggestions.

Appendix F

List of publications

Journals

1. **L. Heryawan**, P.H. Khotimah, O. Sugiyama, G. Yamamoto, L.H.O. Santos, A.E. Pramono, K. Okamoto and T. Kuroda. "Toward design of an agent-based writing support system for the SOAP note: A content analysis of the video-based survey". In: Advanced Biomedical Engineering, 9: 146–153, 2020. DOI: 10.14326/abe.9.146

2. **L. Heryawan**, O. Sugiyama, G. Yamamoto, P.H. Khotimah, L.H.O. Santos, K. Okamoto and T. Kuroda. "A Detection of Informal Abbreviations from Free Text Medical Notes Using Deep Learning". In: European Journal for Biomedical Informatics, 16(1): 29-37, 2020. DOI: 10.24105/ejbi.2020.16.1.29

International Conferences

1. **L. Heryawan**, P.H. Khotimah, G. Yamamoto, O. Sugiyama, S. Hiragi, K. Okamoto and T. Kuroda. "Agent-based Completion for Collecting Medical Note Parameters". In: HAI '19: Proceedings of the 7th International Conference on Human-Agent Interaction (pp. 244-246), Sept. 2019 DOI: 10.1145/3349537.3352780

Bibliography

[1] Travis B Murdoch and Allan S Detsky. The inevitable application of big data to health care. *Jama*, 309(13):1351–1352, 2013. ISSN 0098-7484.

[2] Richard Hillestad, James Bigelow, Anthony Bower, Federico Girosi, Robin Meili, Richard Scoville, and Roger Taylor. Can electronic medical record systems transform health care? Potential health benefits, savings, and costs. *Health affairs*, 24 (5):1103–1117, 2005. ISSN 0278-2715.

[3] Peter B Jensen, Lars J Jensen, and Søren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6):395–405, 2012. ISSN 1471-0064.

[4] Nicole Gray Weiskopf and Chunhua Weng. Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *Journal of the American Medical Informatics Association*, 20(1):144–151, 2013. ISSN 1067-5027.

[5] Yichuan Wang, LeeAnn Kung, and Terry Anthony Byrd. Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological Forecasting and Social Change*, 126:3–13, 2018.

[6] Susan Cameron and Imani Turtle-Song. Learning to write case notes using the soap format. *Journal of Counseling & Development*, 80(3):286–292, 2002.

[7] Inke Mathauer and Friedrich Wittenbecher. Drg-based payment systems in low- and middle-income countries: implementation experiences and challenges. Technical report, World Health Organization Geneva, 2012.

[8] Jason P Van Batavia, Dana A Weiss, Christopher J Long, Julian Madison, Gus McCarthy, Natalie Plachter, and Stephen A Zderic. Using structured data entry systems in the electronic medical record to collect clinical data for quality and research: Can we efficiently serve multiple needs for complex patients with spina bifida? *Journal of pediatric rehabilitation medicine*, 11(4):303–309, 2018.

[9] Anjum Razzaque and Akram Jalal-Karim. Conceptual healthcare knowledge management model for adaptability and interoperability of ehr. In *European, Mediterranean & Middle Eastern Conference on Information Systems*, pages 12–13, 2010.

[10] Ruth A Bush, Cynthia Kuelbs, Julie Ryu, Wen Jiang, and George Chiang. Structured data entry in the electronic medical record: perspectives of pediatric specialty physicians and surgeons. *Journal of medical systems*, 41(5):75, 2017.

[11] Huibert J Tange, Arie Hasman, Pieter F de Vries Robbé, and Harry C Schouten. Medical narratives in electronic medical records. *International journal of medical informatics*, 46(1):7–29, 1997.

[12] Yasushi Matsumura, Shigeki Kuwata, Yuichiro Yamamoto, Kazunori Izumi, Yasushi Okada, Michihiro Hazumi, Sachiko Yoshimoto, Takahiro Mineno, Munetoshi Nagahama, Ayumi Fujii, et al. Template-based data entry for general description in medical records and data transfer to data warehouse for analysis. *Studies in health technology and informatics*, 129(1):412, 2007.

[13] Bryan A Wilbanks, Eta S Berner, Gregory L Alexander, Andres Azuero, Patricia A Patrician, and Jacqueline A Moss. The effect of data-entry template design and anesthesia provider workload on documentation accuracy, documentation efficiency, and user-satisfaction. *International journal of medical informatics*, 118: 29–35, 2018.

[14] Kory Kreimeyer, Matthew Foster, Abhishek Pandey, Nina Arya, Gwendolyn Halford, Sandra F Jones, Richard Forshee, Mark Walderhaug, and Taxiarchis Botsis. Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review. *Journal of biomedical informatics*, 73: 14–29, 2017.

[15] N Lance Downing, David W Bates, and Christopher A Longhurst. Physician burnout in the electronic health record era: are we ignoring the real cause? *Annals of internal medicine*, 169(1):50–51, 2018.

[16] Jack T Marchewka and Kurt Kostiwa. An application of the utaut model for understanding student perceptions using course management software. *Communications of the IIMA*, 7(2):10, 2007.

[17] Michelle O'Daniel and Alan H Rosenstein. Professional communication and team collaboration. In *Patient safety and quality: An evidence-based handbook for nurses*. Agency for Healthcare Research and Quality (US), 2008.

[18] Tom Delbanco, Jan Walker, Jonathan D Darer, Joann G Elmore, Henry J Feldman, Suzanne G Leveille, James D Ralston, Stephen E Ross, Elisabeth Vodicka, and

Valerie D Weber. Open notes: doctors and patients signing on. *Annals of internal medicine*, 153(2):121–125, 2010.

[19] Judith S Olson, Dakuo Wang, Gary M Olson, and Jingwen Zhang. How people write together now: Beginning the investigation with advanced undergraduates in a project course. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(1):1–40, 2017.

[20] Jeremy Birnholtz, Stephanie Steinhardt, and Antonella Pavese. Write here, write now! an experimental study of group maintenance in collaborative writing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 961–970, 2013.

[21] Avner Caspi and Ina Blau. Collaboration and psychological ownership: How does the tension between the two influence perceived learning? *Social Psychology of Education*, 14(2):283–298, 2011.

[22] Leslie A Lenert. Toward medical documentation that enhances situational awareness learning. In *AMIA Annual Symposium Proceedings*, volume 2016, page 763. American Medical Informatics Association, 2016.

[23] Daniel Garrett. Tapping into the value of health data through secondary use: as electronic health records (ehrs) proliferate across the nation, an important new opportunity awaits healthcare organizations that can find meaningful commercial uses for the data contained in their ehr systems. *Healthcare Financial Management*, 64(2):76–84, 2010.

[24] Rainu Kaushal, Kaveh G Shojania, and David W Bates. Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review. *Archives of internal medicine*, 163(12):1409–1416, 2003.

[25] Deborah D Nelson. Copying and pasting patient treatment notes. *AMA Journal of Ethics*, 13(3):144–147, 2011.

[26] Ahmad Tubaishat. The effect of electronic health records on patient safety: a qualitative exploratory study. *Informatics for Health and Social Care*, 44(1):79–91, 2019.

[27] Thomas H Payne, W David Alonso, J Andrew Markiel, Kevin Lybarger, and Andrew A White. Using voice to create hospital progress notes: description of a mobile application and supporting system integrated with a commercial electronic health record. *Journal of biomedical informatics*, 77:91–96, 2018.

[28] Juan C Quiroz, Liliana Laranjo, Ahmet Baki Kocaballi, Shlomo Berkovsky, Dana Rezazadegan, and Enrico Coiera. Challenges of developing a digital scribe to reduce clinical documentation burden. *NPJ digital medicine*, 2(1):1–6, 2019.

[29] Ronilda Lacson and Regina Barzilay. Automatic processing of spoken dialogue in the home hemodialysis domain. In *AMIA Annual Symposium Proceedings*, volume 2005, page 420. American Medical Informatics Association, 2005.

[30] Ruth Reátegui and Sylvie Ratté. Comparison of metamap and ctakes for entity extraction in clinical notes. *BMC medical informatics and decision making*, 18(3): 74, 2018.

[31] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 649–658, 2015.

[32] Peter J Liu. Learning to write notes in electronic health records. *arXiv preprint arXiv:1808.02622*, 2018.

[33] Tyler Baldwin and Joyce Chai. Autonomous self-assessment of autocorrections: exploring text message dialogues. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 710–719, 2012.

[34] Stephen B Johnson, Suzanne Bakken, Daniel Dine, Sookyung Hyun, Eneida Mendonça, Frances Morrison, Tiffani Bright, Tielman Van Vleck, Jesse Wrenn, and Peter Stetson. An electronic health record based on structured narrative. *Journal of the American Medical Informatics Association*, 15(1):54–64, 2008.

[35] Hiroshi Takeda, Yasushi Matsumura, Takeo Okada, Shigeki Kuwata, and Michitoshi Inoue. A japanese approach to establish an electronic patient record system in an intelligent hospital. *International journal of medical informatics*, 49(1):45–52, 1998.

[36] Sri Kusumadewi, Chanifah Indah Ratnasari, and Linda Rosita. Natural language parsing of patient complaints in indonesian language. In *2015 International Conference on Science and Technology (TICST)*, pages 292–297. IEEE, 2015.

[37] Reinhard Busse, Alexander Geissler, Ain Aaviksoo, Francesc Cots, Unto Häkkinen, Conrad Kobel, Céu Mateus, Zeynep Or, Jacqueline O'Reilly, Lisbeth Serdén, et al. Diagnosis related groups in europe: moving towards transparency, efficiency, and quality in hospitals? *Bmj*, 346:f3197, 2013.

[38] SM Aljunid, SM Hamzah, SA Mutalib, AM Nur, N Shafie, and S Sulong. The unu-cbgs: development and deployment of a real international open source casemix grouper for resource challenged countries. *BMC Health Services Research*, 11(Suppl 1):A4, 2011.

[39] S Trent Rosenbloom, Joshua C Denny, Hua Xu, Nancy Lorenzi, William W Stead, and Kevin B Johnson. Data from clinical notes: a perspective on the tension between structure and flexible documentation. *Journal of the American Medical Informatics Association*, 18(2):181–186, 2011.

[40] Sharon Mulvehill, Gregory Schneider, Cassie Murphy Cullen, Shelley Roaten, Barbara Foster, and Anne Porter. Template-guided versus undirected written medical documentation: a prospective, randomized trial in a family medicine residency clinic. *The Journal of the American Board of Family Practice*, 18(6):464–469, 2005.

[41] Colleen Stukenberg. *Successful collaboration in healthcare: A guide for physicians, nurses and clinical documentation specialists*. CRC Press, 2010.

[42] Nicholas Epley, Adam Waytz, and John T Cacioppo. On seeing human: a three-factor theory of anthropomorphism. *Psychological review*, 114(4):864, 2007.

[43] Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 21–35. Springer, 1996.

[44] Kyo-Joong Oh, Dongkun Lee, Byungsoo Ko, and Ho-Jin Choi. A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation. In *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, pages 371–375. IEEE, 2017.

[45] Diane L Adams, Helen Norman, and Valentine J Burroughs. Addressing medical coding and billing part ii: a strategy for achieving compliance. a risk management approach for reducing coding and billing errors. *Journal of the National Medical Association*, 94(6):430, 2002.

[46] C Douglas Phillips and Bruce J Hillman. Coding and reimbursement issues for the radiologist. *Radiology*, 220(1):7–11, 2001.

[47] Danielle Wasserman and Moshe Herskovitz. Epileptic vs psychogenic nonepileptic seizures: a video-based survey. *Epilepsy & Behavior*, 73:42–45, 2017.

[48] Charles P Smith. Content analysis and narrative analysis. 2000.

[49] Marek Fuchs and Frederik Funke. Video web survey. results of an experimental comparison with a text-based web survey. In *Challenges of a changing world. Proceedings of the Fifth International Conference of the Association for Survey Computing*, pages 63–80. Association for Survey Computing Berkeley, 2007.

[50] Jenna Butz, David Brick, Laurie A Rinehart-Thompson, Melanie Brodnik, Amanda M Agnew, and Emily S Patterson. Differences in coder and physician perspectives on the transition to icd-10-cm/pcs: A survey study. *Health Policy and Technology*, 5(3):251–259, 2016.

[51] Karen L Tang, Kelsey Lucyk, and Hude Quan. Coder perspectives on physician-related barriers to producing high-quality administrative data: a qualitative study. *CMAJ open*, 5(3):E617, 2017.

[52] Kimberly Nehls, Brandy D Smith, and Holly A Schneider. Video-conferencing interviews in qualitative research. In *Enhancing qualitative and mixed methods research with technology*, pages 140–157. IGI Global, 2015.

[53] Molli R Grossman, Deanah Kim Zak, and Elizabeth M Zelinski. Mobile apps for caregivers of older adults: Quantitative content analysis. *JMIR mHealth and uHealth*, 6(7):e162, 2018.

[54] Liviu P Lefter, Stuart R Walker, Fleur Dewhurst, and RWL Turner. An audit of operative notes: facts and ways to improve. *ANZ journal of surgery*, 78(9): 800–802, 2008.

[55] Randolph C Barrows Jr, M Busuioc, and Carol Friedman. Limited parsing of notational text visit notes: ad-hoc vs. nlp approaches. In *Proceedings of the AMIA Symposium*, page 51. American Medical Informatics Association, 2000.

[56] Larry E Smith. Spread of english and issues of intelligibility. *The other tongue: English across cultures*, 2:75–90, 1992.

[57] Sylvia Miranda Carneiro, Herica Silva Dutra, Fernanda Mazzoni da Costa, Simone Emerich Mendes, and Cristina Arreguy-Sena. Use of abbreviations in the nursing records of a teaching hospital. *Revista da Rede de Enfermagem do Nordeste*, 17(2):208–216, 2016.

[58] PR Hunt, H Hackman, G Berenholz, Loreta McKeown, Lindsay Davis, and V Ozonoff. Completeness and accuracy of international classification of disease (icd) external cause of injury codes in emergency department electronic data. *Injury Prevention*, 13(6):422–425, 2007.

[59] David C Goodman and Elliott S Fisher. Physician workforce crisis? wrong diagnosis, wrong prescription. *New England Journal of Medicine*, 358(16):1658–1661, 2008.

[60] Hakan Hasbey Koyuncu, Faruk Yencilek, Bilal Eryildirim, and Kemal Sarica. Family history in stone disease: how important is it for the onset of the disease and the incidence of recurrence? *Urological research*, 38(2):105–109, 2010.

[61] Wensheng Wu, Weiyi Meng, Weifeng Su, Guangyou Zhou, and Yao-Yi Chiang. Q2p: discovering query templates via autocompletion. *ACM Transactions on the Web (TWEB)*, 10(2):1–29, 2016.

[62] Deborah L Kasman. When is medical treatment futile? *Journal of general internal medicine*, 19(10):1053–1056, 2004.

[63] Gerald F Riley. Administrative and claims records as sources of health care cost data. *Medical care*, pages S51–S55, 2009.

[64] Roman Vaculín and Roman Neruda. Autonomous behavior of computational agents. In *Adaptive and Natural Computing Algorithms*, pages 514–517. Springer, 2005.

[65] WILFRIED Lepuschitz. 'self-reconfigurable manufacturing control based on ontology-driven automation agents. *Technische Universität Wien, PhD Thesis*, 2018.

[66] Brent Smith and Greg Linden. Two decades of recommender systems at amazon. com. *Ieee internet computing*, 21(3):12–18, 2017.

[67] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[68] Erwan Moreau, François Yvon, and Olivier Cappé. Robust similarity measures for named entities matching. 2008.

[69] Eric L Grogan, Theodore Speroff, Stephen A Deppen, Christianne L Roumie, Tom A Elasy, Robert S Dittus, S Trent Rosenbloom, and Michael D Holzman. Improving documentation of patient acuity level using a progress note template. *Journal of the American College of Surgeons*, 199(3):468–475, 2004.

[70] Felicia Ng. *Am I There Yet?: Probing the Effects of Goal Progress Feedback on Cognitive Motivation*. PhD thesis, Princeton University, 2015.

[71] SA Zafirah, Amrizal Muhammad Nur, Sharifa Ezat Wan Puteh, and Syed Mohamed Aljunid. Potential loss of revenue due to errors in clinical coding during

the implementation of the malaysia diagnosis related group (my-drg®) casemix system in a teaching hospital in malaysia. *BMC health services research*, 18(1):38, 2018.

[72] Jennifer Oates. Use of skype in interviews: The impact of the medium in a study of mental health nurses. *Nurse researcher*, 22(4), 2015.

[73] Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke Van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49, 2015.

[74] Sue E Bowman. Coordination of snomed-ct and icd-10: getting the most out of electronic health record systems. *Coordination of SNOMED-CT and ICD-10: Getting the Most out of Electronic Health Record Systems/AHIMA, American Health Information Management Association*, 2005.

[75] Aleksandra Sarcevic. " who's scribing?" documenting patient encounter during trauma resuscitation. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1899–1908, 2010.

[76] Hamish Fraser, Paul Biondich, Deshen Moodley, Sharon Choi, Burke Mamlin, and Peter Szolovits. Implementing electronic medical record systems in developing countries. *Journal of Innovation in Health Informatics*, 13(2):83–95, 2005.

[77] Vojtech Huser, Chandan Sastry, Matthew Breymaier, Asma Idriss, and James J Cimino. Standardizing data exchange for clinical research protocols and case report forms: An assessment of the suitability of the clinical data interchange standards consortium (cdisc) operational data model (odm). *Journal of biomedical informatics*, 57:88–99, 2015.

[78] Moritz Lehne, Julian Sass, Andrea Essenwanger, Josef Schepers, and Sylvia Thun. Why digital medicine depends on interoperability. *NPJ Digital Medicine*, 2(1):1–5, 2019.

[79] Miriam Reisman. Ehrs: the challenge of making electronic data usable and interoperable. *Pharmacy and Therapeutics*, 42(9):572, 2017.

[80] Jean Marie Rodrigues, Stefan Schulz, Alan L Rector, Kent A Spackman, Bedirhan Üstün, Christopher G Chute, Vincenzo Della Mea, Jane Millar, and Kristina Brand Persson. Sharing ontology between icd 11 and snomed ct will enable seamless reuse and semantic interoperability. In *MedInfo*, pages 343–346, 2013.

[81] MK Ross, Wei Wei, and L Ohno-Machado. "big data" and the electronic health record. *Yearbook of medical informatics*, 23(01):97–104, 2014.

[82] Joanna E Sheppard, Laura CE Weidner, Saher Zakai, Simon Fountain-Polley, and Judith Williams. Ambiguous abbreviations: an audit of abbreviations in paediatric note keeping. *Archives of disease in childhood*, 93(3):204–206, 2008.

[83] Kathleen E Walsh and Jerry H Gurwitz. Medical abbreviations: writing little and communicating less. *Archives of disease in childhood*, 93(10):816–817, 2008.

[84] Ivy Fenton Kuhn. Abbreviations and acronyms in healthcare: when shorter isn't sweeter. *Pediatric nursing*, 33(5), 2007.

[85] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[86] Sungrim Moon, Bridget McInnes, and Genevieve B Melton. Challenges and practical approaches with word sense disambiguation of acronyms and abbreviations in the clinical domain. *Healthcare informatics research*, 21(1):35–42, 2015.

[87] Antonio Jimeno Yepes. Word embeddings and recurrent neural networks based on long-short term memory nodes in supervised biomedical word sense disambiguation. *Journal of biomedical informatics*, 73:137–147, 2017.

[88] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[89] Zhen Yang, Matthias Dehmer, Olli Yli-Harja, and Frank Emmert-Streib. Combining deep learning with token selection for patient phenotyping from electronic health records. *Scientific Reports*, 10(1):1–18, 2020.

[90] Chen Lyu, Bo Chen, Yafeng Ren, and Donghong Ji. Long short-term memory rnn for biomedical named entity recognition. *BMC bioinformatics*, 18(1):462, 2017.

[91] Marc-Antoine Rondeau and Yi Su. Lstm-based neurocrfs for named entity recognition. In *INTERSPEECH*, pages 665–669, 2016.

[92] Yonghui Wu, S Trent Rosenbloom, Joshua C Denny, Randolph A Miller, Subramani Mani, Dario A Giuse, and Hua Xu. Detecting abbreviations in discharge summaries using machine learning methods. In *AMIA Annual Symposium Proceedings*, volume 2011, page 1541. American Medical Informatics Association, 2011.

[93] Lana Yeganova, Donald C Comeau, and W John Wilbur. Identifying abbreviation definitions machine learning with naturally labeled data. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 499–505. IEEE, 2010.

[94] Arzoo Katiyar and Claire Cardie. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, 2018.

[95] Lin Yao, Hong Liu, Yi Liu, Xinxin Li, and Muhammad Waqas Anwar. Biomedical named entity recognition based on deep neutral network. *Int. J. Hybrid Inf. Technol*, 8(8):279–288, 2015.

[96] Arjun Magge, Abeed Sarker, Azadeh Nikfarjam, and Graciela Gonzalez-Hernandez. Comment on:"deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts". *Journal of the American Medical Informatics Association*, 26(6):577–579, 2019.

[97] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[98] Mahendra Sahare and Hitesh Gupta. A review of multi-class classification for imbalanced data. *International Journal of Advanced Computer Research*, 2(3): 160, 2012.

[99] Hao Li, Yanyan Shen, and Yanmin Zhu. Stock price prediction using attention-based multi-input lstm. In *Asian Conference on Machine Learning*, pages 454–469, 2018.

[100] Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno. Speaker diarization with lstm. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5239–5243. IEEE, 2018.

[101] Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. A unified tagging solution: Bidirectional lstm recurrent neural network with word embedding. *arXiv preprint arXiv:1511.00215*, 2015.

[102] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks*, pages 799–804. Springer, 2005.

[103] Patrick Ruch, Robert Baud, and Antoine Geissbühler. Evaluating and reducing the effect of data corruption when applying bag of words approaches to medical records. *International Journal of Medical Informatics*, 67(1-3):75–83, 2002.

[104] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

[105] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[106] Neal Lewis, Daniel Gruhl, and Hui Yang. Extracting family history diagnosis from clinical texts. In *BICoB*, pages 128–133, 2011.

[107] Shinji Kobayashi, Naoto Kume, Takahiro Nakahara, and Hiroyuki Yoshihara. Designing clinical concept models for a nationwide electronic health records system for japan. *European Journal of Biomedical Informatics*, 2018.

[108] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[109] Di Zhao, Jian Wang, Hongfei Lin, Zhihao Yang, and Yijia Zhang. Extracting drug–drug interactions with hybrid bidirectional gated recurrent unit and graph convolutional network. *Journal of Biomedical Informatics*, 99:103295, 2019.

[110] Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. Overview of biocreative ii gene mention recognition. *Genome biology*, 9(S2):S2, 2008.

[111] Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*, 2016.

[112] Gondy Leroy, Hsinchun Chen, and Jesse D Martinez. A shallow parser based on closed-class words to capture relations in biomedical text. *Journal of biomedical Informatics*, 36(3):145–158, 2003.

[113] Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 747–754, 2017.

[114] Roman Klinger and Christoph M Friedrich. User's choice of precision and recall in named entity recognition. In *Proceedings of the International Conference RANLP-2009*, pages 192–196, 2009.

[115] Hadaiq Rolis Sanabila and Wisnu Jatmiko. Ensemble learning on large scale financial imbalanced data. In *2018 International Workshop on Big Data and Information Security (IWBIS)*, pages 93–98. IEEE, 2018.

[116] Clayton A Turner, Alexander D Jacobs, Cassios K Marques, James C Oates, Diane L Kamen, Paul E Anderson, and Jihad S Obeid. Word2vec inversion and traditional text classifiers for phenotyping lupus. *BMC medical informatics and decision making*, 17(1):1–11, 2017.

[117] Hongfang Liu, Stephen T Wu, Dingcheng Li, Siddhartha Jonnalagadda, Sunghwan Sohn, Kavishwar Wagholikar, Peter J Haug, Stanley M Huff, and Christopher G Chute. Towards a semantic lexicon for clinical natural language processing. In *AMIA Annual Symposium Proceedings*, volume 2012, page 568. American Medical Informatics Association, 2012.

[118] Chaker Jebari, Manuel Jesús Cobo, and Enrique Herrera-Viedma. A new approach for implicit citation extraction. In *International conference on intelligent data engineering and automated learning*, pages 121–129. Springer, 2018.

[119] Hengliang Wang, Yuan Li, Chenfei Zhao, Yi Zhang, and Kedian Mu. A meta-strategy enhancement for network embedding. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1720–1723. IEEE, 2019.

[120] Parinya Sanguansat. Paragraph2vec-based sentiment analysis on social media for business in thailand. In *2016 8th International Conference on Knowledge and Smart Technology (KST)*, pages 175–178. IEEE, 2016.

[121] VS Anoop and S Asharaf. Distributional semantic phrase clustering and conceptualization using probabilistic knowledgebase. In *International Conference on Next Generation Computing Technologies*, pages 526–534. Springer, 2017.

[122] Robert Bill, Serguei Pakhomov, Elizabeth S Chen, Tamara J Winden, Elizabeth W Carter, and Genevieve B Melton. Automated extraction of family history information from clinical notes. In *AMIA Annual Symposium Proceedings*, volume 2014, page 1709. American Medical Informatics Association, 2014.

[123] Elizabeth W Carter, Indra Neil Sarkar, Genevieve B Melton, and Elizabeth S Chen. Representation of drug use in biomedical standards, clinical text, and research measures. In *AMIA Annual Symposium Proceedings*, volume 2015, page 376. American Medical Informatics Association, 2015.

[124] Linda Flower and John R Hayes. A cognitive process theory of writing. *College composition and communication*, 32(4):365–387, 1981.

[125] Melanie L Balestra. Electronic health records: patient care and ethical and legal implications for nurse practitioners. *The Journal for Nurse Practitioners*, 13(2): 105–111, 2017.

[126] Anders Green, Helge Huttenrauch, and K Severinson Eklundh. Applying the wizard-of-oz framework to cooperative service discovery and configuration. In *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No. 04TH8759)*, pages 575–580. IEEE, 2004.

[127] Steven Dow, Blair MacIntyre, Jaemin Lee, Christopher Oezbek, Jay David Bolter, and Maribeth Gandy. Wizard of oz support throughout an iterative design process. *IEEE Pervasive Computing*, 4(4):18–26, 2005.

[128] David Kurlander. Persona: An architecture for animated agent interfaces.

[129] Tae Kyun Kim. T test as a parametric statistic. *Korean journal of anesthesiology*, 68(6):540, 2015.

[130] SPSS Tutorials. Paired samples t test. *Retrieved from libguides. library. kent. edu/SPSS/PairedSamplestTest.*

[131] Amy Hennington and Brian D Janz. Information systems and healthcare xvi: physician adoption of electronic medical records: applying the utaut model in a healthcare context. *Communications of the Association for Information Systems*, 19(1):5, 2007.

[132] Ralph L Keeney. Value-focused thinking: Identifying decision opportunities and creating alternatives. *European Journal of operational research*, 92(3):537–549, 1996.

[133] Padmanabhan Ramnarayan, Graham C Roberts, Michael Coren, Vasantha Nanduri, Amanda Tomlinson, Paul M Taylor, Jeremy C Wyatt, and Joseph F Britto. Assessment of the potential impact of a reminder system on the reduction of diagnostic errors: a quasi-experimental study. *BMC Medical Informatics and Decision Making*, 6(1):22, 2006.

[134] Ziv Bar-Yossef and Naama Kraus. Context-sensitive query auto-completion. In *Proceedings of the 20th international conference on World wide web*, pages 107–116, 2011.

[135] Ohoud Alharbi, Ahmed Sabbir Arif, Wolfgang Stuerzlinger, Mark D Dunlop, and Andreas Komninos. Wisetype: a tablet keyboard with color-coded visualization and various editing options for error correction. *Graphics Interface 2019*, 2019.

[136] Jillian Madison. *Damn You, Autocorrect!* Random House, 2012.

[137] Mia Xu Chen, Benjamin N Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M Dai, Zhifeng Chen, et al. Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2287–2295, 2019.

[138] Nicola Wood. Autocorrect awareness: Categorizing autocorrect changes and measuring authorial perceptions. 2014.

[139] John Sweller and Paul Chandler. Evidence for cognitive load theory. *Cognition and instruction*, 8(4):351–362, 1991.

[140] John Sweller, Paul Ayres, and Slava Kalyuga. The redundancy effect. In *Cognitive load theory*, pages 141–154. Springer, 2011.

[141] Kimberly J O'malley, Karon F Cook, Matt D Price, Kimberly Raiford Wildes, John F Hurdle, and Carol M Ashton. Measuring diagnoses: Icd code accuracy. *Health services research*, 40(5p2):1620–1639, 2005.

[142] Martin McKee. Routine data: a resource for clinical audit? *Quality in Health Care*, 2(2):104, 1993.

[143] Barbara A Lopez. Establishing a cdi program: How one organization leveraged nursing and coding skills to improve clinical data. *Journal of AHIMA*, 81(7):58–59, 2010.

[144] Thomas Payne. Improving clinical documentation in an emr world: one health system used a three-step process to integrate a clinical documentation improvement program with its new electronic medical record system, with significant results. *Healthcare Financial Management*, 64(2):70–75, 2010.

[145] Andy Weeger and Heiko Gewald. Acceptance and use of electronic medical records: An exploratory study of hospital physicians' salient beliefs about hit systems. *Health Systems*, 4(1):64–81, 2015.

[146] Princely Ifinedo. Technology acceptance by health professionals in canada: An analysis with a modified utaut model. In *2012 45th Hawaii international conference on system sciences*, pages 2937–2946. IEEE, 2012.

[147] Ju-Ling Hsiao and Rai-Fu Chen. Critical factors influencing physicians' intention to use computerized clinical practice guidelines: an integrative model of activity theory and the technology acceptance model. *BMC medical informatics and decision making*, 16(1):3, 2015.

[148] Tinuke M Fapohunda. Towards effective team building in the workplace. *International Journal of Education and Research*, 1(4):1–12, 2013.

[149] Mirjam Körner, Markus A Wirtz, Jürgen Bengel, and Anja S Göritz. Relationship of organizational culture, teamwork and job satisfaction in interprofessional teams. *BMC health services research*, 15(1):243, 2015.

[150] Michelle X Zhou, Gloria Mark, Jingyi Li, and Huahai Yang. Trusting virtual agents: The effect of personality. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 9(2-3):1–36, 2019.

[151] Byron Reeves and Clifford Nass. How people treat computers, television, and new media like real people and places, 1996.

[152] Jesse Fox, Sun Joo Ahn, Joris H Janssen, Leo Yeykelis, Kathryn Y Segovia, and Jeremy N Bailenson. Avatars versus agents: a meta-analysis quantifying the effect of agency on social influence. *Human–Computer Interaction*, 30(5):401–432, 2015.

[153] Masahiko Nakamura. Current status of electronic medical recording in japan and issues involved. *Japan Medical Association Journal*, 49(2):70, 2006.

[154] Robert H Miller and Ida Sim. Physicians' use of electronic medical records: barriers and solutions. *Health affairs*, 23(2):116–126, 2004.