Utrecht University

Master Thesis

VIVE: An LLM-based approach to identifying and extracting context-specific personal values from text

Author:
Raoul Brigola

First supervisor:
Asst. Prof. Davide Dell'Anna

In cooperation with:
Netherlands Red Cross

Second supervisor:
Prof. Pinar Yolum

A master thesis submitted in fulfillment of the requirements for the degree of Master of Science in Artificial Intelligence

Department of Information and Computing Science

June 27, 2024

Declaration of Authorship

I, Raoul Brigola, declare that this thesis titled, "VIVE: An LLM-based approach to identifying and extracting context-specific personal values from text" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a master's degree at Utrecht University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Utrecht 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: June 27, 2024

UTRECHT UNIVERSITY

Abstract

Faculty of Science
Department of Information and Computing Science

Master of Science

VIVE: An LLM-based approach to identifying and extracting context-specific personal values from text

by Raoul BRIGOLA

Personal values are referenced in natural language text through subtle cues that indicate a person's priorities and beliefs. Understanding these values requires advanced natural language understanding to correctly interpret subtleties and nuances. Existing research on extracting personal values from text often utilizes methods that lack sufficient natural language understanding and do not consider the context of a text. In this study, we present VIVE (*Value Identification and Value Extraction*), a novel end-to-end method for the identification and extraction of context-specific personal values from natural language text. VIVE leverages a hybrid intelligence approach to identify which values are particularly important in a given context (*Value Identification*) and utilizes the natural language understanding capabilities of state-of-the-art large language models (LLMs) to extract the identified values from text (*Value Extraction*). To evaluate VIVE, we conduct a case study with the Netherlands Red Cross in which we elicit the requirements of humanitarian organizations with regard to processing feedback data from humanitarian programs. We apply VIVE to the context of a humanitarian program within which the Red Cross collects chat messages from Telegram groups, written by Ukrainian refugees or internally displaced people. VIVE is used to 1) identify a set of context-specific personal values for the data set of Ukrainian Telegram messages and 2) extract these values from the messages. We evaluate the accuracy, precision, recall, and F1 score of the value extraction and we conduct a user study with Red Cross analysts to evaluate the usefulness of VIVE. We find that large language models can accurately extract personal values from text and outperform a traditional dictionary-based approach. Based on this result, we make a comparison of three state-of-the-art LLMs and find no significant difference in their accuracy for value extraction. Furthermore, we show that representing personal values not only through names but also with natural language descriptions significantly improves the accuracy of value extraction and we present a value representation format that is suitable for an LLM-based value extraction.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial Intelligence (AI) is considered a multipurpose technology. One of its many use cases is to support individuals and organizations in making well-informed decisions by providing critical insights. This becomes especially valuable when dealing with vast amounts of data. AI systems are able to skim data quickly and extract and summarize relevant information. This accelerates decision-making but also ensures that key insights are not overlooked in the process. As AI continues to evolve, it becomes increasingly crucial for data driven decision-making.

For example, AI can support the decision-making of humanitarian organizations by analyzing feedback data collected from humanitarian projects [5][19]. Humanitarian organizations, like the International Red Cross and Red Crescent Movement (ICRC), implement a variety of programs to alleviate human suffering and protect the lives and dignity of people around the world. All data that is collected within the scope of such programs is regarded as feedback data. From flood prediction systems, to automated damage assessment, to the optimization of resource distribution, humanitarian programs increasingly rely on both real-time and historical data. Along with the increasing importance of feedback data, the impact of AI on humanitarian programs continues to grow.

The rapid development of AI brings forth a variety of risks and challenges with regard to its societal impact and ethical implications. Current AI systems often exhibit biases and AI-supported decision-making processes lack transparency. In light of these developments, governments and academia have started to recognize the need for *ethical AI* [65][8]. This is shown by the increasing number of AI safety and ethics guidelines that are put into place. The Institute of Electrical and Electronics Engineers (IEEE) has started an initiative for ethically aligned design, with the declared objective to "provide guidelines/procedures/standards to prioritize human well-being in the forthcoming evolution of artificial intelligence and autonomous systems" [65]. Similarly, a collective of AI researchers has mapped out an Artificial Intelligence Research Agenda for the Netherlands [8].

The term *ethical AI* refers to artificial intelligence systems that adhere to such guidelines. The intention of ethical AI is to ensure that AI applications align with humanitarian principles and values. Stuart Russell calls this the *value alignment problem* [60][59]. Russell states that the larger goal of AI development should be defined as developing "intelligence provably aligned with human values". He thereby emphasizes the relevancy of human values for decision-making. The increased research effort on ethical AI [71] has resulted in the proposal of various computational frameworks for human values and value sensitive design [46][22]. In the context of humanitarian projects, ethical AI and AI value alignment become particularly important due to the inherent purpose of these programs to protect the rights of people, especially people that are in a vulnerable situation. In fact, AI-influenced decisions can often directly impact the lives and well-being of people.

In many cases, feedback data from humanitarian programs contains natural language text. For example, messages that are sent to a helpline of a humanitarian organization. With the goal of value alignment in mind, AI systems used for decision-making based on natural language text data should take into consideration the human values communicated through natural language. To this day, the natural language understanding of machines lacks a deep understanding of human personal values. Artificial Intelligence is not yet able to fully grasp the concept of a personal value, which contributes to the aforementioned *value alignment problem* [13]. The value alignment problem is viewed as the challenge of making sure that AI systems understand and follow human values, so that they act in ways that are beneficial to humans. This raises various research questions: How to represent human personal values computationally? How to define a reference to a value in natural language in a machine-translatable way? Overall, how to artificially replicate a human-like understanding of the abstract concept of personal values?

The goal of this study is to develop an artificial intelligence tool that supports human decision-making, for example, the decision-making of humanitarian aid workers, while taking these questions into account.

1.1 Personal Values in Natural Language Text

Personal values play an important role in the way humans communicate. Following a widespread view from social science, personal values are the foundational beliefs that influence a person's decision-making and overall behavior [61][55]. Therefore, identifying the personal values behind a statement can also reveal a person's intentions. Understanding what personal values are communicated in a text can have several benefits. It allows a deeper analysis of the text, as it might uncover hidden needs and desires of the author. While often personal values are not explicitly mentioned, many statements and arguments are motivated by them.

In its most common usage, the term *personal value* refers to very broad, fundamental concepts, like *freedom*, *security*, or *sustainability*. Personal values are often considered trans-contextual, meaning they apply in any given situation. However, depending on the context, such broad terms typically have many possible interpretations. For example, in the political context of an oppressed minority, the personal value *freedom* might stand for freedom of speech or freedom of religion. In the context of the relationship between teenagers and their parents, *freedom* is associated with autonomy and the ability to make independent choices. In yet another context of a software developer, *freedom* can refer to the principles of open-source and the ability to use software without restrictions.

It also inherently depends on the context, which personal values someone deems relevant and what their prioritization is. For example, in the context of a startup company working in agriculture, environmental sustainability is likely a very relevant value. On the other hand, for a profit-oriented trading company environmental sustainability might be lower ranked in the prioritization of values. Furthermore, certain values are not even applicable to some contexts. For example, in a restaurant context, people can be assumed to value *culinary quality*. On the other hand, in an astrophysics context this value has little meaning. Therefore, when analyzing a statement, it is important to consider the context in which the statement is made and the personal values that are relevant in that context. Section 1.5 provides formal definitions for the terms *personal value* and *context-specific value*.

As humans, we have learned from a young age to understand what personal values are communicated to us. Our understanding of natural language allows us to detect references to personal values in text or speech. For example, from the statement "As a school teacher, it is important to me that the individual needs of all students are taken into account", we can infer that the person that made the statement values *inclusivity*. Furthermore, we understand that inclusivity is a relevant personal value in the context of *schooling*. Several cognitive processes lead to this understanding of personal values. In this work, we simplify them into the tasks of *value identification* and *value extraction*. Generally speaking, value identification can be viewed as the task of determining what personal values are important to a person or a group of people, and value extraction is the task of detecting what personal values are referenced in a given piece of natural language text. Section 1.5.1 elaborates on what we understand under these two terms and provides formal definitions for them.

1.2 Large Language Models and Personal Values

Two central assumptions of this study are first, that extracting personal values from natural language text requires an advanced level of natural language understanding, and second, that large language models (LLMs) possess sufficient natural language understanding to extract personal values. Traditional natural language processing (NLP) methods can often not sufficiently comprehend the nuances and complexities of natural language, to extract personal values. For example, the Bag-of-Words (BoW) method [51] represents a text by counting the frequency of words in it, without considering the order or context of the words. While BoW can be useful for simple text classification tasks or keyword extraction, it falls short when dealing with more abstract concepts, like personal values. In contrast, LLMs are trained on vast amounts of natural language texts. Recent advances in the field of LLMs show that they possess extensive natural language understanding [40][73]. A crucial advantage of LLMs compared to traditional NLP methods is their large context window and the associated ability to understand the context of words in a sentence and the context of sentences or paragraphs in a text. Furthermore testing LLMs on benchmarks for natural language understanding [40][73] shows that they are often able to understand idiomatic expressions and cultural references. These abilities are beneficial when extracting personal values from natural language text, especially when the values are not explicitly stated, like in the above example of the personal value *inclusivity* being referenced in the sentence "As a school teacher, it is important to me that the individual needs of all students are taken into account".

1.3 Methodology

In this study, we follow the Design Science Methodology (DSM) for Information Systems and Software Engineering from Roel J. Wieringa [75]. We use this methodology because it is well-tested and specifically designed for IT systems. Additionally, the iterative nature of the DSM allows continuous refinement and improvement, which makes it suitable for the complex requirements of humanitarian aid work.

The design cycle proposed by Wieringa consists of three steps: A problem investigation, a treatment design, and a treatment validation. The *problem investigation* refers to the systematic analysis of the challenges that are present in the context of the research objective. These challenges should be addressed by the methodology.

In this study, the problem investigation pertains to the questions of what requirements humanitarian organizations have for processing feedback data and how AI can support the involved decision-making processes. Section 1.4 describes in detail the problem investigation we conducted for this study.

The *treatment design* refers to the development of a solution to the identified challenges. The treatment design typically is the design of an artifact. In this study, the artifact is VIVE - a novel method for the identification and extraction of personal values from natural language text. VIVE is described in detail in section 3. According to Wieringa, the *treatment* itself is "the interaction between the artifact and the problem context". Meaning, how the method is applied to solve the problem.

Finally, the *treatment validation* confirms how well the designed artifact addresses the requirements identified in the problem investigation. In this study, the treatment validation is a comprehensive evaluation of VIVE.

The design cycle is part of a larger, iterative process - the engineering cycle. The engineering cycle describes a generic problem-solving process and includes a treatment implementation and an evaluation of the implementation. A treatment implementation, which is a real-world application based on the designed treatment, exceeds the scope of this study. Therefore, we apply the design cycle, as shown in figure 1.1, for this project.

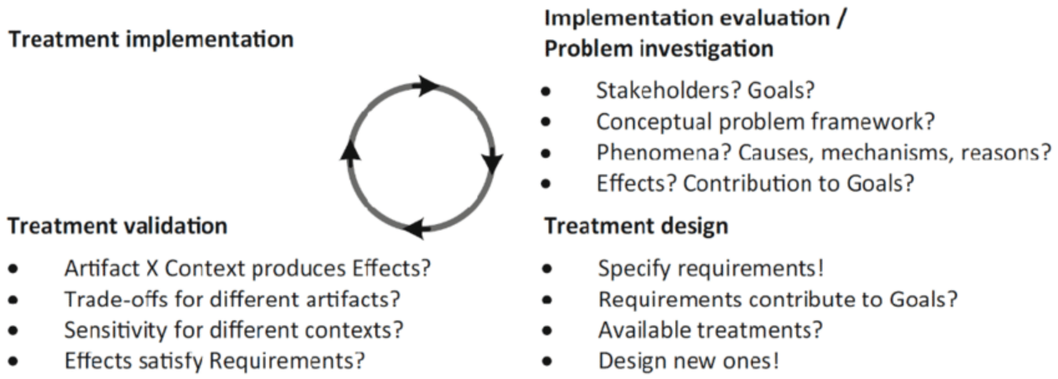

Treatment implementation
Implementation evaluation /
Problem investigation
• Stakeholders? Goals?
• Conceptual problem framework?
• Phenomena? Causes, mechanisms, reasons?
• Effects? Contribution to Goals?
Treatment validation
• Artifact X Context produces Effects?
• Trade-offs for different artifacts?
• Sensitivity for different contexts?
• Effects satisfy Requirements?
Treatment design
• Specify requirements!
• Requirements contribute to Goals?
• Available treatments?
• Design new ones!

FIGURE 1.1: Design Cycle by Wieringa (taken from [75])

1.4 Problem Investigation

The problem investigation conducted for this study essentially consists of two parts: Firstly, we elicit requirements directly from stakeholders, namely analysts of the Netherlands Red Cross. Secondly, we review related literature to identify available methods and their limitations.

1.4.1 Case Study: The Netherlands Red Cross

This research project is done in collaboration with the Netherlands Red Cross. The Netherlands Red Cross implements various data-driven humanitarian programs [1].

[1] More information on 510, an initiative of the Netherlands Red Cross, and its digital humanitarian programs, can be found under:
`https://510.global/about-us/`

Within the project *Social Media Listening* (SML)[2], the Netherlands Red Cross collects chat messages from public Telegram groups. The selected Telegram groups are used by Ukrainian refugees or internally displaced people (IDPs) in Ukraine, to communicate and help each other. People send messages to exchange information and ask questions about a variety of topics, from transportation to health care to applications for subsidies.

Following the Design Science Methodology by Wieringa et al. [75] (see section 1.3), we conducted a problem investigation with analysts of the Netherlands Red Cross. The conducted problem investigation consists of a focus group meeting and a total of 16 interactive discussions with Red Cross analysts. To understand the needs of stakeholders from the Red Cross with regard to processing feedback data, we held a focus group meeting with three Red Cross analysts [3] in accordance with the "Practical Guide to Focus-Group Research", by Rosanna L. Breen [11]. Through the focus group meeting, we aimed to answer the following questions: What is a Red Cross analyst looking for in feedback data? How can an automated analysis of feedback data support a Red Cross analyst? As a guiding example, we looked at the processing of the collected Telegram chat messages. The term *Red Cross analyst* refers to a Red Cross employee who uses feedback data to implement and improve humanitarian projects. We summarize the results of the focus group in the following requirements:

Requirement 1 (Automation)**.** *A tool to automatically extract information relevant to humanitarian programs from feedback data, as the amount of feedback data makes manual processing impractical.*

Humanitarian organizations, like the Netherlands Red Cross, would like to use feedback data to optimize the use of their resources. For example, during the focus group meeting, participants agreed to the following statement: "We use the Ukraine feedback data to more efficiently distribute resources from the Red Cross." However, the amount of feedback data that is collected is vast. For example, the data collected within SML contains thousands of messages per day. To accurately extract relevant information from the data, automized analysis tools are necessary.

Requirement 2 (Understanding the experience)**.** *A tool that analyzes feedback data to gain insights into individuals' needs and experiences, enabling the design of humanitarian programs with personalized support and improved communication.*

During the focus group meeting, participants emphasized the benefits of understanding the personal values that are particularly important to people in a humanitarian crisis situation. With regard to the SML project, participants made the following statements: "If we knew the motivations and needs and values of a person that writes the message, we could probably [...] cluster them better and present them better and probably try to get better help for them", "The essence of it is understanding the experience, needs, and challenges that people affected by disaster or crisis have and making sure that the Red Cross activities and programs are addressing those needs and experiences."

[2] Information about the SML project can be found under:
`https://510.global/product/sml/`,
`https://github.com/rodekruis/social-media-listening/tree/master`

[3] The presentation slides from the focus group meeting can be found in this project's GitHub repository: `https://github.com/brigoraoul/VIVE`, under *Netherlands Red Cross Case Study/Focus Group/FocusGroupSlides.pptx*

Requirement 3 (Actionable output). *A tool to translate information from feedback data into concrete actions.*

Many of the tools that are currently available to humanitarian organizations focus on reporting and the summarization of data. However, these tools could be more valuable if they would provide concrete steps to improve an individual's situation. This requirement is supported by statements of the focus group participants, like the following: "So how do you turn this feedback data into actionable information that people can actually act upon in the reality that they're working in?", "In an ideal world, it's actually not only creating that overview and identifying trends, but it's actually getting back to the individual that has a certain experience or need or help request and helping them either with support that the Red Cross can give or by referring them exactly that."

1.4.2 Literature Review

The literature review we conducted as part of our problem investigation focuses on existing methods to identify and extract personal values from natural language text. The full literature review is reported in section 2. Here, we summarize the main shortcomings of existing methods in the following two limitations. To address them, we formulate a research contribution per identified limitation.

Limitation 1 (LLM-based method). *Despite the advanced natural language understanding of large language models (LLMs), there is no method for value extraction that makes use of them.*

Research Contribution 1. *We present a novel LLM-based method for value extraction, utilizing the superior natural language understanding of LLMs compared to traditional natural language processing methods. We thereby address the lack of an LLM-based value extraction method.*

Limitation 2 (End-to-end method). *There is no method that provides an end-to-end solution for the identification and extraction of context-specific values, making it necessary to use multiple tools for the identification and extraction of context-specific values and ensuring their compatibility.*

Research Contribution 2. *We present an end-to-end solution for the identification and extraction of context-specific values. In this study, we refer to an end-to-end solution as a method that includes all steps involved to obtain a natural language data set labeled with context-specific values from a raw data set. An end-to-end method can be applied easily to new contexts and data sets, streamlining the process and guaranteeing compatibility of the individual modules.*

1.5 Research Objective

As mentioned above, the goal of this study is to develop an artificial intelligence tool that supports human decision-making, for example, the decision-making of humanitarian aid workers, while accounting for human values. By doing so, we aim to answer the main research question described in section 1.5.2. We build on existing methods (see section 2 and particularly section 2.4), but do not aim to make a comparison of them. Instead, our proposed method - VIVE - addresses the shortcomings of existing approaches (see section 2.7). These shortcomings are identified through our literature review and are summarized in limitation 1 and 2. Research contribution 1 addresses limitation 1 and research contribution 2 addresses limitation 2.

1.5.1 Definitions and Notations

In the following, we formally introduce a number of terms and notations, based on which we specify the research questions of this study in section 1.5.2.

Definition 1 (Personal Value)**.** *A personal value is a fundamental belief or principle that determines a person's attitudes, behaviors, and decision-making in life.*

Definition 2 (Context)**.** *A context encompasses all environmental, social, and cultural elements that contribute to a certain situation. Furthermore, the context of a situation comprises all involved actors and their actions with all their consequences. A context can also have a spatial or a temporal scope, meaning it can be restricted to a certain area or time period.*

To describe a situation, it is inevitable to describe elements of the context. At the same time, due to the vastness of its characteristics, it is often not possible to define a context completely. We argue that the personal values that a person holds are inherently dependent on the circumstances in which the person finds themselves. In this study, we aim to identify context-specific personal values (see 1.5.2). Given the above definition of a context, the personal values that someone deems particularly important in a certain context are themselves part of that context.

Definition 3 (Context-specific Value)**.** *A context-specific value is a personal value that people deem particularly important within a given context.*

By their nature, context-specific values are unique to a person and their priority can change over time. A central assumption of this work is that any human actor a holds a set of context-specific values $V_{a,c}$ for any context C. For the hypothetical case that an actor a is completely indifferent towards a context C, $V_{a,c}$ is the empty set. Given a set of actors A, we call $V_{c,A}$, the set of collective context-specific values of all actors $a \in A$, that is $V_{c,A} = \bigcup_{a=1}^{|A|} V_{c,a}$.

Definition 4 (Value Identification)**.** *Given a context C and a set of human actors A, value identification is the task of determining $V_{c,A}$, the set of collective context-specific values of all actors $a \in A$.*

As mentioned above, *value identification* can be viewed as the task of determining what personal values are important to a person or a group of people. It has to be differentiated between the identification of general, trans-contextual values and context-specific values. Following definition 3 for *context-specific values*, definition 4 constrains the task of *value identification* to the identification of context-specific values. This generally requires an understanding of the context and a definition of its scope. The term *actor* is used here as a synonym for *person*.

Definition 5 (Natural language data set)**.** *A natural language data set is a collection of texts, written in natural language.*

A text $d \in D_c$ in a natural language data set D_c can be anything from a word to a sentence to an actual text. The **author** of d is the person who wrote the text. A non-empty natural language data set D_c has a set of authors A_D, with at least one author ($|A_D| \geq 1$). A natural language data set D_c **pertains** to the context C in which the data was generated or collected.

Definition 6 (Value Extraction)**.** *Given a context C, a pertaining natural language data set D_c with a set of authors A_D that hold a set of context-specific values $V_{c,A}$, value extraction is the task of determining for each data point $d \in D_c$, the set of referenced values $V_d \subset V_{c,A}$.*

In this work, we refer to *value extraction* as the task of detecting what personal values are referenced in a given piece of natural language text. Principally, value extraction is a *multi-label task*, as multiple personal values can be referenced in a given text. A *value extraction method* generally refers to a method that can be used for the task of value extraction. The way the term *value extraction* is used here, presumes *value identification*. Meaning, that to perform value extraction it is necessary to already have identified a (finite) set of context-specific values. The literature review in section 2 explores existing definitions of what a *reference to a value* is and existing approaches to automated value extraction.

Definition 7 (Single-label Value Extraction)**.** *Given a context C, a pertaining natural language data set D_c with a set of authors A_D that hold a set of context-specific values $V_{c,A}$, value extraction is the task of determining for each data point $d \in D_c$, the primarily referenced value $v_d \in V_{c,A}$.*

In contrast to definition 6, definition 7 specifies value extraction as a *single-label task*. Meaning, that for a given text d, exactly one or no value can be extracted. We view the *primarily referenced value v_d* of a text d as the personal value that is referenced the strongest.

1.5.2 Research Questions

Main research question:

> *How to automatically extract context-specific personal values from natural language text?*

The main research question pertains to the definition 6 for value extraction. It essentially poses the question of how to achieve the task that is defined as *value extraction*. Values refer to the personal values that the author of a natural language text aims to convey. These are regarded as context-specific values, following definition 3. The following sub-research questions address specific aspects of the main research question. Answering them can be viewed as a prerequisite for a complete answer to the main research question.

Sub-research question 1:

> *How to identify context-specific values?*

This sub-research question pertains to the above definition of value identification (definition 4). Section 1.5.1 mentions that value identification is presumed by value extraction. In simple terms, it needs to be clear what values are relevant before value extraction can be performed. Therefore, this work addresses the problem of value identification as a sub-research question.

Sub-research question 2:

> *How to represent personal values computationally?*

Because personal values are often abstract concepts (see section 2.1), it is not trivial how to formally represent them. A simple representation of a personal value could be a natural language *word*. More complex computational representations, like word embeddings, are conceivable. Related work largely agrees that it is necessary to find a computational representation of values to develop a method for automated value extraction [46][22].

Sub-research question 3:

> *How to determine for a piece of text $d \in D_c$ and a context-specific value $v \in V_c$, whether v is referenced in d? More specifically, what is a function*
>
> $$f : d \in D_c, v \in V_c \mapsto \begin{cases} 1 & ; \quad v\, is\, referenced\, in\, d \\ 0 & ; \quad otherwise \end{cases}$$
>
> *that indicates for arbitrary $d \in D_c$ and $v \in V_c$, whether v is referenced in d?*

Following definition 6, the main research question pertains to the question of which values are referenced in a given piece of text. To answer it, it is necessary to have a way to decide for each value in a given set of context-specific values individually whether it is referenced or not. Different definitions of a *reference to a value* can be found in the literature (see section 2.4). The definition of a reference to a value inherently depends on the value representation. We call a method that provides an answer to sub-research question 3 a *value extraction source*.

Sub-research question 4:

> *"What is the accuracy, precision, recall, and usefulness of the proposed method for value extraction?"*

Evaluating the performance of our proposed method indicates how well it achieves the objectives outlined in this section. Furthermore, it allows a comparison to existing methods and therefore an approximation of the contribution to the field. In this work, we evaluate the proposed method through standard performance measures, namely, accuracy, precision, and recall, and measures that are calculated based on them, like the F1-score. Besides that, we assess the usefulness of the method, as a more qualitative performance measure. There is no widely agreed on definition of the term *usefulness*. In this work, we regard usefulness as a measure of how practical and valuable the output of a system is to the user. For example, in the context of the Red Cross case study (see section 1.4), a measure for usefulness becomes a measure of how helpful the extracted values are for the decision-making of Red Cross analysts. This can be assessed via a user study.

1.6 Outline of this Study

The sections of this document are structured to convey a comprehensive and clear impression of our research methodology when read in order.

- Section 2 summarizes existing research related to personal values, value identification, and value extraction. It highlights the main limitations and shortcomings of existing research. Additionally, background information on some parts of our method is provided.

- Section 3 describes VIVE, our proposed method for value identification and value extraction, on a conceptual level.

- Section 4 describes how we apply the VIVE pipeline to address the requirements elicited from the Netherlands Red Cross (see 1.4.1) and the identified limitations of related literature (see 1.4.2). It includes a description of the used data set and constitutes the method section of this study.

- Section 5 reports how we evaluate our instantiation of VIVE through a series of experiments. This section includes our experimental method, setup, and all obtained results. Furthermore, section 5 poses three experimental research questions and provides a discussion of them based on the results.

- Section 6 reflects on the broader research questions of this study (see 1.5.2) and discusses the implications of our findings, before giving an outlook on future work.

- Finally, section 7 summarizes the main findings and contributions of this study.

Chapter 2

Background and Related Work

The following presents an overview of the background and related work, examining prior research on *value identification* (subsection 2.3), *value representations* (subsection 2.6) and *value extraction* (subsection 2.4). It follows the definition 4 and 6 of the terms *value identification* and *value extraction* from the Introduction, and elaborates on definitions 1 and 2, for the terms *personal value* (2.1) and *context* (2.2). The approaches from related work are briefly explained in their methodology and limitations are highlighted. Lastly, this section includes a review of *value-based computing systems* (2.5) and a summary of the different *value representations* that exist in the literature (2.6).

2.1 Personal Values

While this work is not considered research on personal values per se, for the goals of this work it is important to have an understanding of what a personal value is. Definition 1, from section 1.5.1, is very broad and therefore allows interpretation. Most commonly, personal values are regarded as abstract concepts that motivate a person's behavior and can be interpreted as desirable goals of a person [57]. Schwartz et al. [64] define the concept of a value through five features: "Values (1) are concepts or beliefs, (2) pertain to desirable end states or behaviors, (3) transcend specific situations, (4) guide selection or evaluation of behavior and events, and (5) are ordered by relative importance". Friedman et al. [22] say "a value refers to what a person or group of people consider important in life" and mention that this broad definition allows values like *children*, *morning tea* or *a walk in the woods*. Unlike the common definition of a personal value as a broad, trans-situational belief or goal, in this study we allow more specific, context-dependent, and possibly temporary goals to be personal values. As definition 3 states, we consider anything that is of importance to a person in a given situation or context a context-specific personal value.

Lilach Sagiv et al. give an overview of the research on personal values, mostly stemming from social sciences [61]. They mention the existence of a hierarchy as an important aspect of personal values. Each person has an individual value hierarchy that provides guidance for behavior and decision-making. The higher a value is located in the hierarchy, the more likely a person acts according to it. In case of a conflict, a value higher up trumps a value at a lower stage of the hierarchy. While a value hierarchy is unique to a person, there are similarities in the value hierarchy of most people. While value hierarchies are not the central research objective of this study, we consider them in the design and implementation of our method and experiments.

2.1.1 Generic Value Lists

Many attempts to formalize personal values assume the existence of universal values. More specifically, they make the assumption that there exists a set of universal values that is relevant to people across all cultures, and applicable to every context. These approaches reject the idea that people have different sets of values, but typically acknowledge that people have individual value hierarchies or prioritizations.

Most often cited is the Schwartz Theory of Basic Values [64] [63]. Schwartz identifies 10 universal human values. Figure 2.1 shows the theoretical arrangement of these 10 Schwartz values.

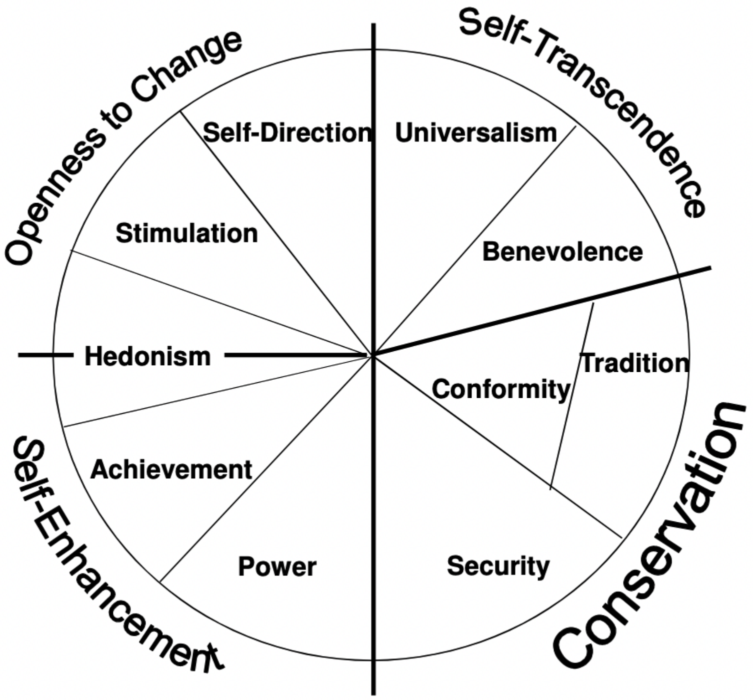


FIGURE 2.1: Schwartz Value Dimensions (taken from [63])

While figure 2.1 suggests some relations between different values, Schwartz argues that all 10 values are distinct in their motivation. This means they cannot be substituted for each other and therefore represent a value system of 10 dimensions. The Schwartz Value Survey provides a method to measure these 10 values. Typically, the survey includes a number of statements that reflect the different values. Participants are asked to indicate how strongly they agree with the statements, which makes an individual value-ranking possible.

Similarly, the Rokeach Value Survey [56] seeks to derive a ranking of 18 terminal values and 18 instrumental values. Terminal values are defined as goals that a person would like to achieve, like *wisdom*, *inner harmony*, or *an exciting life* and Rokeach acknowledges that they might differ across cultures.

The concept of personal values is closely related to the concept of moral values. A number of theories exist that provide a generic list of moral values. Most notably, Moral Foundations Theory (MFT) identifies the 6 moral foundations that determine a person's actions and decisions: care/harm, fairness/cheating, loyalty/betrayal, authority/subversion, sanctity/degradation, and liberty/oppression [26]. The theory argues that human moral values have evolved through evolution and are therefore universal.

Generic value lists argue that it is not necessary to identify personal values based on the context. They solve the problem of value identification by simply applying the same set of values to every context. Consequently, generic value lists take the individual characteristics of a situation into account only to a limited extent. Furthermore, a number of works question the universality claim of the values. For example, de Wet et al. [74] show through an empirical study that the prioritization of the Schwartz values is indeed context-dependent. In their experiment, students completed the Schwartz Portrait Value Questionnaire one time without a particular context in mind, a second time with their family/home as the context, and a third time with their university as the context. They found that depending on the context some values play a very little role and can therefore be assumed to not be relevant in that context.

2.2 The term *Context*

In this study, we use the term context according to definition 2, from section 1.5.1. This definition is the result of a review of various definitions of the term *context* from literature. The Cambridge Dictionary defines a context as "The influences and events related to a particular event or situation" [36]. Such a definition implies a number of characteristics that a context can have. In the article "An ontology-based context model in intelligent environments" Gu et al. [27] state "By context, we refer to any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object." In the article “Understanding and Using Context”, Anind K. Dey [17] provides an even more inclusive definition, by saying: "If a piece of information can be used to characterize the situation of a participant in an interaction, then that information is context." Similar definitions can be found in context-modeling literature [68].

In natural language processing, the term *context* often refers to the lexical context of a word, for example, the words or sentences that precede or follow a word in a text [32] or dependencies within a dependency tree [24]. This is not how the term *context* is used in this work.

2.2.1 Context-aware Systems

In many software engineering applications, it is desirable that a software agent can consider the context to which it is applied when taking an action. By understanding the context-specific circumstances and user preferences, a context-aware software agent can align its actions with the given situation. This improves the adaptability and effectiveness of its decisions.

Baldauf et al. [3] conduct a survey on context-aware systems and summarize common elements and architecture principles. They mention a *context model* as an essential element of a context-aware system. A context model is a way to formalize a context so that it can be processed by a computer program. Examples for context models are sensory nodes or context component frameworks, like CORTEX [6]. Another integral part of context-aware systems is a context processing module. The survey from Baldauf et al. [3] includes context-processing approaches based on relational data models, ontologies and object-oriented programming. Many context-aware systems make use of historical context-specific data. Baldauf et al. also emphasize the importance of security and privacy in context-aware systems, as sensitive context-specific data must be protected.

2.3 Value Identification

The introduction defines *value identification* as the task of determining the personal values that someone holds and, possibly, ranking them by their importance. This section gives an overview of the proposed methods for value identification from the literature. In contrast to universal values, context-specific values can be viewed as particularly relevant values for a given context. While generic value lists (section 2.1.1) normally consist of a finite number of values, the number of conceivable context-specific values can be infinite. For practical reasons, the approaches that are reviewed in the following aim to derive a finite set of values. Some works follow a hybrid intelligence approach. The reviewed literature in this section is primarily related to sub-research question 1 from section 1.5.2.

2.3.1 Axies

Figure 2.2 shows an overview of *Axies*, a methodology to identify context-specific values, proposed by Liscio et al. [39].

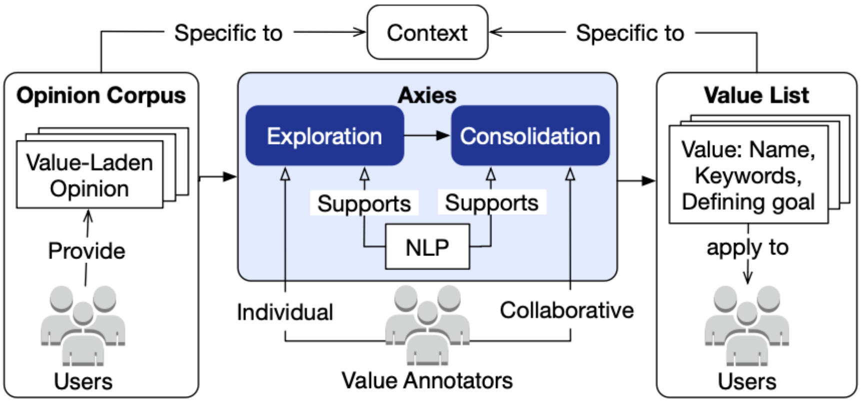


FIGURE 2.2: Axies Workflow (taken from [39])

Axies takes as input an "opinion corpus", a collection of texts in which users express opinions that are motivated by the values the hold. Axies outputs a list of values that, which Liscio et al. claim is specific to the context of the opinion corpus and applicable to the users that produced the opinion corpus. The method follows a hybrid intelligence approach, using Farthest First Traversal (FFT) [4] to select instances from the data for annotation. FFT starts by picking a random data point. The next data point that is picked is the farthest away from the starting point and gets added to the set of traversed data points. This requires a measure to determine how far two data points are away from one another. In the context of Axies, this measure is a distance measure in the sentence embedding space. Axies uses Sentence-BERT, from Reimers et al. [54], to locate a data point in the sentence embedding space.

The basic idea is that sentences or texts that are semantically similar lie closer in the sentence embedding space.

The annotation of the context-specific personal values is done by a small group of annotators that each produce their own list of values. This process is called *exploration*. In a second step, called *consolidation*, the individual value lists are merged to obtain one final list. The process is based on collaboration amongst the annotators but is guided by Axies, which provides the annotators with a fixed set of actions to take at any step of the collaboration. Section 3 describes in detail how and with which modifications we incorporate Axies into the VIVE pipeline.

The focus of the Axies method is to utilize NLP and active learning techniques to find context-specific values. Liscio et al. [39] show that the use of these techniques improves the context-specificity of the obtained values compared to other techniques. Seriously judging the context-specificity of a derived set of values requires a measure of how specific a value is to a context.

2.3.2 Meaning Extraction Method

Boyd et al. [10] propose a method for context-specific value identification based on the Meaning Extraction Method (MEM) [14]. The study emphasizes the benefits of context-specific values and shows that they yield better results than a generic value list, namely the Schwartz values, when predicting people's behavior. One of their main research objectives is to investigate whether values should be identified through "traditional self-reports" or through the analysis of natural language. The approach that Boyd et al. present, aims to derive "meaningful words" - which represent the author's values - from text by solely looking at features of the text. The method does not follow a hybrid intelligence approach and relies on the assumption that all information that is necessary to derive context-specific values can be found in text.

The meaning extraction method (MEM) was originally thought of as a way to identify categories of self-reflection from texts, in which people describe themselves [14]. In the article "An Approach to Evaluate Content Patterns From Large-Scale Language Data", [41] Markowitz shows that MEM can be generalized to different contexts, making it a generic method to identify themes from text. MEM preprocesses the data by removing function words and infrequent words. In a second step, MEM reduces the dimensionality - the number of words - of the data through Principal Component Analysis (PCA). Only words are kept that achieve a value above a certain threshold for each component. Markowitz mentions that the optimal threshold is dependent on the data set. MEM allows a tuning of how broad the identified themes are, by regulating the total number of identified themes - the higher the number of themes, the more specific they are. Markowitz makes use of the Meaning Extraction Helper [9], a MEM-based, automated tool to derive meaningful words from text.

2.3.3 Other Methods

Natural language processing (NLP) literature provides a number of methods that can potentially be used for value identification.

As a generic tool to identify categories from text, Fast et al. propose "Empath" [18]. Empath relies on a collection of 1.8 billion learned word embeddings that allow measures of similarity between words. Figure 2.3 shows all steps of the Empath method.

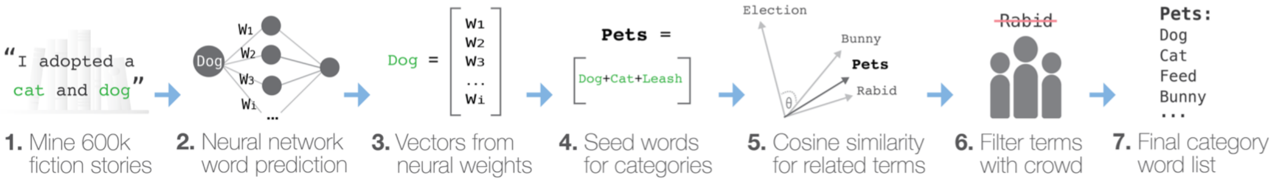

"I adopted a cat and dog"
Dog
W1
W2
W3
Wi
...
Dog =
W1
W2
W3
...
Wi
Pets =
Dog+Cat+Leash
Election
Bunny
Pets
Rabid
θ
~~Rabid~~
Pets:
Dog
Cat
Feed
Bunny
...
1. Mine 600k fiction stories
2. Neural network word prediction
3. Vectors from neural weights
4. Seed words for categories
5. Cosine similarity for related terms
6. Filter terms with crowd
7. Final category word list

FIGURE 2.3: Empath Workflow (taken from [18])

Provided with only a few "seed terms", Empath uses its word embeddings to define a category and find related words. Additionally, it includes 200 built-in categories. The methodology follows a hybrid intelligence approach. It validates a newly created category through a crowd-powered rating process of the words that form the category. The motivation for this step is to avoid unrelated words that were accidentally assigned to a category. The fundamental methodology of Empath can be applied to personal values, if a suitable similarity measure is found. However, it is not specifically build for value identification. Similar techniques for dictionary categories have long existed in text analysis literature [47] [67]. A weakness of fully automized methods, like MEM or Empath, is that they do not take into account the nuances of human contextual understanding. However, we believe that these nuances and subtleties are crucial for the performance of a value identification method.

Several works perform value identification based on commonsense estimation of context-specific values. This is often possible for contexts in which a generally accepted set of personal values exists. For example, in the context of a hotel booking site, it is a fair assumption that a customer values *location*, *cleanliness*, *quality of the facilities*, etc. Chang et al. [12] make such an assumption when analyzing hotel reviews from TripAdvisor. Similarly, Yamaguchi et al. [78] use a commonsense estimation of personal values in the context of movie reviews.

Witesman et al. [76] suggest an empirical approach to identify context-specific value hierarchies and demonstrate how it can support decision-making in five contexts related to public policy making. In their pilot study, context-specific values are identified through questionnaires. In a second step, individual context-specific value hierarchies are identified by presenting "decision scenarios" to the participants and asking them about their value preferences. Identifying a value hierarchy with the method proposed by Witesman et al. requires significant effort because it essentially includes a user study. Furthermore, Witesman et al. focus on the collective context-specific values of all involved actors. In contrast, in this study, we propose a hybrid intelligence method that, depending on the individual user and data point, allows different value hierarchies within the same context.

2.4 Value Extraction

Given a pre-defined set of personal values, the introduction defines *value extraction* (see definition 6) as the task of detecting which of these values are referenced in a given piece of text. However, some of the literature that is reviewed in the following does not differentiate between the identification and extraction of values and instead regards them as one step. Principally, the existing approaches extract personal values from natural language text can be categorized into machine learning methods (see section 2.4.1) and dictionary-based methods (see section 2.4.2). The definition of a *reference to a value in text* differs between these two categories. Dictionary-based approaches generally store pairs of value-names and a list of keywords that represent the value [31] [50]. Given such a dictionary, a reference to a value is simply defined as an occurrence of one of the value's keywords in text. For example, if *wound* and *healing* are considered keywords for the value *health*, the sentence *I don't know why the wound on my leg does not heal.* is considered to reference the value *health*. On the other hand, machine learning approaches, like Teenrstra et al. [70], typically utilize less intuitive definitions of what a reference to a value is. The exact definitions depend on the text features that a machine learning model uses. The subsection 2.4.3 describes an LLM inference-based approach to value extraction.

2.4.1 Machine-learning Methods

Asprino et al. [2] propose two unsupervised methods to detect latent moral content in natural language text: a zero-shot learning approach and a frame-based approach. Both methods are evaluated on a dataset of tweets labeled according to the Moral Foundation Theory (MFT). The zero-shot learning approach uses a pre-trained model to recognize moral values in tweets. The frame-based approach makes use of existing knowledge graphs and semantic web technologies. Building on prior research, Asprino et al. transform tweets into knowledge graphs, which were then analyzed to detect moral values. A limitation of their work is that the models do not take into account any context information surrounding the tweets.

Motger et al. [44] propose "T-FREX: A Transformer-based Feature Extraction Method from Mobile App Reviews", a novel approach for automatically extracting features from mobile app reviews using large language models (LLMs). While T-FREX focuses on feature extraction, its methodology for understanding and analysing text through LLMs can be adapted to the task of value extraction. Figure 2.4 gives an overview of the T-FREX research design.

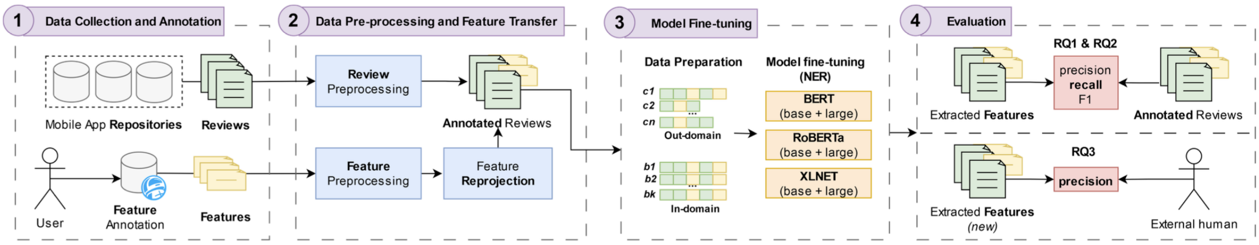


FIGURE 2.4: T-FREX - Transformer-based feature extraction (taken from [44])

Motger et al. gather user-annotated features from a crowdsourced platform and collect a data set of app reviews. Both are pre-processed into a unified format (e.g., CoNLL-U), linking features to relevant keywords in the user reviews. Ensuing, different LLM architectures are fine-tuned on the data set of app reviews. T-FREX's performance is evaluated through token-level classification metrics, like precision, recall, and the F1-score and compared to a baseline method. Overall, T-FREX explores the potential of LLMs to improve feature extraction accuracy and the identification of new features and feature categories.

Based on Moral Foundations Theory (MFT) [26] (see section 2.1.1), Teernstra et al. [70] develop a machine learning approach to extract MFT values from tweets. Their methodology follows a supervised learning approach, as they manually label a small portion of the collected tweets with a single MFT value for training. The methodology does not approach the task as a multi-label problem, meaning it is based on the assumption that each tweet can be assigned exactly one of the MFT values. This poses a limitation, as different values are generally not mutually exclusive. For classification, Teernstra et al. use multinomial naive bayes [35] and maximum entropy models [45]. They conclude that machine learning models can be used for the extraction of values from natural language, but acknowledge that using values from a generic list like MFT poses limitations.

2.4.2 Dictionary-based Methods

Also building on MFT, Hopp et al. develop the "extended Moral Foundations Dictionary" (eMFD), taking a dictionary-based approach to value extraction [31]. The eMFD dictionary is constructed through crowd-sourced annotation of words. Instead of choosing experts for annotation, a large group of people with presumably different backgrounds assign MFT values to words from news articles. This results in a list of keywords for each value - or "moral intuition" as Hopp et al call it. Because a single word can be annotated with multiple values, in contrast to [70], the eMFD approaches value extraction as a multi-label problem. The eMFD is constructed in the context of news articles. Because it is based on a generic value list, it is questionable how well it performs when applied to a different context. Nonetheless, the crowd-sourced annotation methodology can in theory be used to construct a context-specific dictionary for any set of context-specific values.

Ponizovskiy et al. follow a similar dictionary-based approach [50] based on the Schwartz value list [64], with the goal of an "automatic assessment of references to personal values in text". Their method to construct the dictionary relies on a small group of experts to collaboratively construct a list of candidate words for each of the Schwartz values. The candidate words are validated by checking how frequently words that are selected for the same value co-occur in text. Various data sets, like collections of personal blogs, essays, and Facebook updates, are used for this validation. Based on the results, the dictionary with the candidate words is refined, which leads to a dictionary with reduced size. Analogously to [31], the methodology lacks context-specificity, as the Schwartz values are not specific to a certain context.

2.4.3 LLM inference-based Value Extraction

This section provides background information on the functionality of large language models (LLMs). More specifically, it specifies the *LLM inference* function. In simple terms, LLM inference is a function that takes as input a natural language prompt and outputs a natural language text response. We define the inference process formally as a function $LLM_{inference}(X) = O$, where X represents an input sequence $X = (x_1, x_2, \ldots, x_t)$, normally a sequence of words. O represents the output sequence $O = (o_1, o_2, \ldots, o_t)$, also a sequence of words. To the best of our knowledge, at the time of writing, there are no value extraction methods that extract context-specific personal values through LLM inference.

The underlying architecture of LLMs is typically based on a transformer architecture, as originally proposed by Vaswani et al. [72]. To realize the LLM inference function, transformers make use of an attention mechanism, which allows the model to focus on specific parts of the input sequence X when generating each element of the output sequence O. This is achieved by calculating attention scores between each pair of words in the input sequence. The attention scores indicate the importance of each input word for predicting the current output word. In the attention matrix $A \in R^{t \times t}$, A_{ij} denotes the attention score between the i-th and j-th word in the input sequence. The attention scores are typically computed using a scoring function S:

$$S(x_i, x_j) = f(W_q x_i, W_k x_j, W_v x_j)$$

Here, W_q, W_q, and W_q are weight matrices, and f is a non-linear activation function. The attention weights α are then obtained by applying a softmax function over the attention scores:

$$\alpha \in R^{t \times t} = softmax(A)$$

The attention weights represent a probability distribution over all words in the input sequence, indicating how much "attention" the model should pay to each word when generating the current output word. Through multiple layers of attention and feed-forward operations, an LLM is able to construct an output sequence O, while considering the context of the entire input sequence for each word $o \in O$.

This definition of the LLM inference function and its output is based on the fundamental functionality of transformer architectures, as described by Vaswani et al. [72]. Similar definitions can be found in the literature [52].

Comparison to LLM sequence classification A possible LLM alternative to inference-based value extraction is LLM sequence classification. Sequence classification requires training a model to classify input texts, by providing labeled data. Labeled data refers to natural language texts labeled with the context-specific values that they reference. In contrast to the inference approach, the LLM output of a sequence classification task is not a generated text, but a probability distribution over the provided labels. We identify several advantages of using inference for the task of value extraction: 1. Inference makes it possible to retrace the LLM's reasoning behind an extracted value, in the form of an explanation. This is particularly helpful when dealing with abstract concepts like personal values (see section 2.1). 2. Inference does not require the LLM to be retrained when new labels are added. 3. Via prompt engineering, the task of value extraction can be defined as a multi-label task or a single-label task with very minimal effort. With sequence classification, this requires separate training setups.

2.5 Value-based Systems

The term *value-based system* refers to a computing system or technology that considers human personal values in its design and functionality. Since personal values are the focus of this project, it is important to consider existing research on value-based systems, as it contributes to tackling the main research question (see section 1.5.2). The following sections review literature on how to design and develop value-based applications.

2.5.1 Value sensitive Design

Friedman et al. provide an extensive definition of value sensitive design and examine the steps necessary to derive achieve a value-based system [22][21]. More specifically, they identify the necessary conceptual, empirical and technical investigations. A conceptual investigation involves exploring who is affected by the design, how they are affected and what personal values are important. This also includes looking at trade-offs between different values. The purpose of an empirical investigation is to support and validate the conceptual investigation and its assumptions. This can involve a number of user studies, like interviews or questionnaires. Finally, a technical investigation involves all implementational aspects that impact or are dependent on the personal values of users.

In their work, Friedman et al. assume that values that are identified in the design phase of a system hold true over time. The proposed method for value sensitive design does not provide an option to change the design in case the prioritisation of values changes. Pointing out this drawback, van de Poel [48] proposes a value change taxonomy. van de Poel acknowledges that values are context-specific and can change over time and discusses technical features of value-bases systems that "might help to better deal with value change".

2.5.2 Values and Agents

Heidari et al. propose a framework for software agents to make "value-based decisions" [29], based on the Schwartz values [64]. They follow the idea that a universal set of values exists and that individuals are solely different in their value priorities / hierarchies. They acknowledge that value hierarchies are context-specific and that given a certain context some values can be "silent", meaning irrelevant. However, they do not address the argument that a generic value list, like the Schwartz values, are not fine-grained enough to precisely describe individual value hierarchies for a wide range of contexts. At the core of the framework is a mapping from a set of values (Schwartz values) to an importance score (for example in the range [1, 100]). Given a measure for the importance score, a value hierarchy can be constructed. This in turn allows an agent to select between actions and prioritise goals.

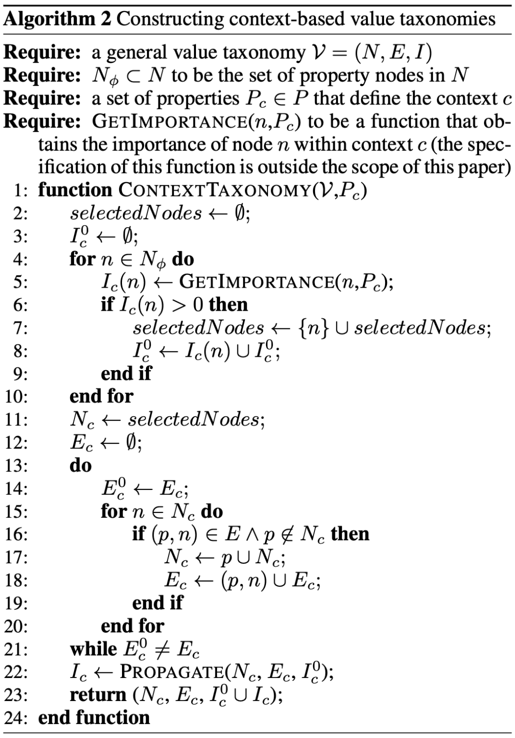
2.6 Note on Value Representation

Across the literature that is reviewed in the preceding sections different ways of representing values are used. This section summarizes different ways of representing values computationally, thereby summarizing literature related to sub-research question 2 (see section 1.5.2).

Most commonly, values have a lexical representations, like a single word or a short description in natural language. The generic value lists presented in section 2.1.1 and the value extraction methods that build on top of them (section 2.4) almost all acknowledge a single word as a valid representation of a value. Lexical representation are intuitive and can be handy for analysis tasks [9]. A disadvantage of is that they cannot be computed, like for example word embeddings.

Dictionary-based approaches to value extraction generally represent a value by its name and a corresponding list of keywords or phrases [31][50]. Ponizovskiy et al. use a vector representation of values, where value in the vector indicates how strongly a word represents the value [50]. A number of unique value representations exist in literature [39][46].

With the field of artificial intelligence and the AI value alignment problem in mind, Osman et al. [46] claim to have proposed the first formal, computational framework of human values. In their definition of values, they acknowledge that the relevancy and meaning of values varies depending on the context and can change over time. The framework represents values through value taxonomies that can be expressed as directed, acyclic graphs. A value taxonomy is not a representation of one value, but rather a network of values where abstract values, like *fairness* are located higher up in the graph and more specific and concrete values can be found in the leaf nodes. A directed edge between two nodes indicates that the value in the parent node is a more general concept than the value in the child node. Finally, to account for context-specificity, the framework includes an importance function that assigns a context-dependent importance value to the nodes of a value taxonomy.



```
Algorithm 2 Constructing context-based value taxonomies
Require: a general value taxonomy $\mathcal{V} = (N, E, I)$
Require: $N_\phi \subset N$ to be the set of property nodes in $N$
Require: a set of properties $P_c \in P$ that define the context $c$
Require: GETIMPORTANCE($n$,$P_c$) to be a function that obtains the importance of node $n$ within context $c$ (the specification of this function is outside the scope of this paper)
function CONTEXTTAXONOMY($\mathcal{V}$,$P_c$)
    $selectedNodes \leftarrow \emptyset$;
    $I_c^0 \leftarrow \emptyset$;
    for $n \in N_\phi$ do
        $I_c(n) \leftarrow$ GETIMPORTANCE($n$,$P_c$);
        if $I_c(n) > 0$ then
            $selectedNodes \leftarrow \{n\} \cup selectedNodes$;
            $I_c^0 \leftarrow I_c(n) \cup I_c^0$;
        end if
    end for
    $N_c \leftarrow selectedNodes$;
    $E_c \leftarrow \emptyset$;
    do
        $E_c^0 \leftarrow E_c$;
        for $n \in N_c$ do
            if $(p, n) \in E \wedge p \notin N_c$ then
                $N_c \leftarrow p \cup N_c$;
                $E_c \leftarrow (p, n) \cup E_c$;
            end if
        end for
    while $E_c^0 \neq E_c$
    $I_c \leftarrow$ PROPAGATE($N_c$, $E_c$, $I_c^0$);
    return $(N_c, E_c, I_c^0 \cup I_c)$;
end function
```

FIGURE 2.5: Constructing context-specific value taxonomies (taken from [46])

2.7 Shortcomings of Existing Research

The above literature review provides an overview of existing research related to this study and highlights the weaknesses of individual methodologies. In this section, we briefly summarize these weaknesses and mention some shortcomings of the current state of research.

As explained in section 2.4, the existing works on value extraction can be categorized into dictionary-based and machine learning approaches. The former has fundamental limitations when trying to extract context-specific values: To the best of our knowledge, all dictionary-based approaches take a generic value list as a basis [31] [50]. There are no methodologies for the construction of a dictionary from a set of context-specific values. Given the variety of conceivable contexts, it seems unrealistic to construct a dictionary that is extensive enough to be applicable to every context. Furthermore, depending on the context, a certain keyword may or may not be a reference to a certain value. Likewise, to the best of our knowledge, all existing machine learning approaches to value extraction use a generic value list [70].

The methodologies of some works on context-specific value identification could be combined with existing value extraction methods [10]. However, there is no end-to-end method for the identification and extraction of context-specific values from text. Also, only a few of the reviewed works explicitly follow a value-sensitive design method.

In this study, we address the limitations of existing works on value identification and value extraction that are summarized as limitations 1 and 2 in section 1.4.2.

Chapter 3

VIVE - Value Identification and Value Extraction

This section presents *VIVE*, our proposed end-to-end method for the identification and extraction of personal values from natural language text data. Proposing such an end-to-end method is stated as research contribution 2 of this study, in section 1.5. Part of VIVE is a value extraction module that uses large language models (LLMs) for the task of value extraction. The use of LLMs for the task of value extraction is described as research contribution 1 in section 1.4.2.

Figure 3.1 graphically displays the modules that VIVE consists of and their order. All rectangles depict a module that includes some functionality, for example, the manipulation of data or the extraction of information from it. All ovals depict the input/output data of the modules to which they are connected. For example, the *value representation* module has the values, as identified by the *value identification* module, as input and it outputs a set of value representations that can be used by the *value extraction* module. In the following, the individual modules and their interplay, as shown in figure 3.1, is referred to as *the VIVE pipeline*. The following subsections explain in detail the functionality of each module on a conceptual level. Section 4 shows how the VIVE pipeline can be instantiated and used in a real-world context. Principally, all modules of the VIVE pipeline can be considered to be of equal importance. However, given the main research question from section 1.5.2, this study puts a particular focus on the value extraction module.

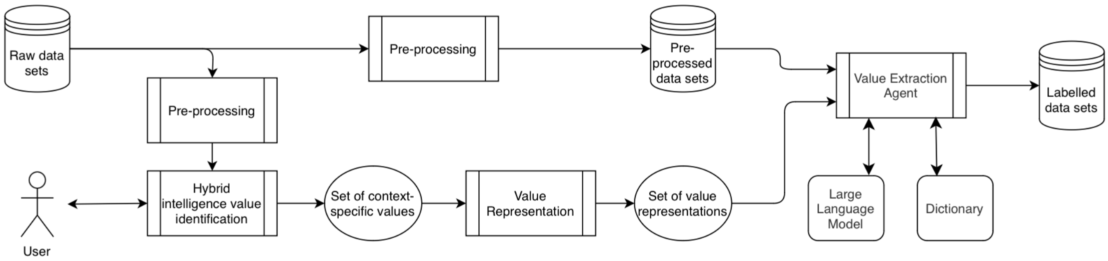


FIGURE 3.1: VIVE pipeline for the identification and extraction of context-specific personal values

3.1 Value Identification

The first module of the pipeline shown in figure 3.1 is the value identification module. The value identification module is responsible for the task of *value identification* as specified in definition 4. In the context of this study, it addresses sub-research question 1 from section 1.5.2: *"How to identify context-specific values?"*

As figure 3.2 shows, the value identification module interacts with the user to process the input data. The goal of this interaction is to combine human and machine intelligence to perform value identification. We identify several benefits of such a hybrid intelligence solution over a completely automated value identification method. 1) Personal values are subjective concepts and by their nature unique to each person (see section 2.1) [61][49]. For example, in the context of social media messages from war refugees, one author of a message might interpret the personal value *security* as being safe from physical harm, while another might interpret it as having a secure financial situation. 2) Following requirement 2 (see section 1.4), we aim to build a tool that given a natural language data set helps to understand the experience of the authors of the data set. Hence, the system needs to understand the context-specific personal values of the authors as well as possible. For these reasons, we deem a hybrid intelligence solution for the value identification suitable.

The value identification module produces a list of context-specific personal values. The format of this list and the way in which values are represented depend on the specific implementation of the value identification module. For example, the value list could be stored as a table in a database, where each value is represented as a word in natural language.

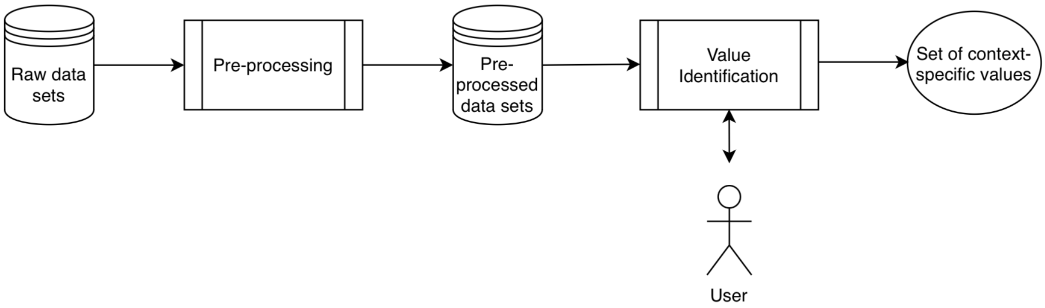


FIGURE 3.2: Section of the VIVE pipeline with the value identification module and the corresponding pre-processing module.

3.1.1 Pre-Processing

Pre-processing of the data is optional for the value identification and depends on the data set at hand. In some cases, it might be possible to use the raw data directly for the value identification. In other cases, pre-processing might include transforming the data into a format that is required for the value identification process. Additionally, a filtering of the data can be appropriate in cases where a large portion of the data is not of interest for the value identification and extraction. For each criterion used for filtering, it should be ensured that no data is filtered out that may contain relevant information for creating a value list that is as complete as possible.

3.2 Value Representation

Section 1.5.2 states the sub-research question 2 of this work as *"How to represent personal values computationally?"*. Simply put, representing personal values computationally means encoding personal values into a format that can be understood and processed by computers. As figure 3.1 shows, value identification precedes value representation in the VIVE pipeline. This means the first step is to identify the context-specific values at hand. In the second step, the identified values can be encoded into the chosen format. Section 2.6 gives an overview of different value representations present in literature. Various categories of value representations are conceivable, amongst others these include lexical representations, like a natural language word or description text, numerical representations, like a score, or structured representations, like a dictionary or ontology. Personal values can evolve over time and the values that are relevant to a given context can change. Therefore, computational value representations should be flexible by design. They should be able to be modified based on new information or new circumstances.

VIVE represents a personal value as a triple <*n, K, D*>, where *n* is the value name, *K* is a list of keywords associated with the value, and *D* is a description of the value. The description *D* is meant to explain what it means to hold a certain value, given the context. All three elements of the VIVE value representation (name, keywords, description) are in natural language. This makes it a convenient format for value extraction based on large language models because the represented values can serve as direct inputs for these models. The inclusion of keywords makes the chosen value representation particularly suitable for the construction of a dictionary. Another benefit of the chosen value representation is that it can be adapted. The keyword list *K* and the description *D* can be extended or changed, allowing a refinement of the value representation over time. This makes it possible to account for changing circumstances, like a change in the relevancy or meaning of the context-specific values. The VIVE value representation is based on a value representation format suggested by Liscio et al. [39].

In the VIVE pipeline, the value representation module functions as a link between the value identification and the value extraction module. As figure 3.3 shows, it takes a set of context-specific values as input, in the format used by the value identification module. It outputs a set of context-specific values that is processable by the value extraction module. The value representation module can be viewed as a translator that transforms the identified values into a suitable format.

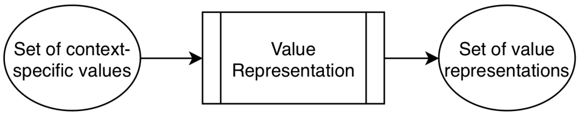


FIGURE 3.3: Section of the VIVE pipeline with the value representation module and its input and output.

3.3 Value Extraction

Following the value representation module in the VIVE pipeline is the value extraction module. The value extraction module handles the task of value extraction as specified in definition 6. It plays an integral role in answering this study's main research question *How to automatically extract context-specific personal values from natural language text?* (see section 1.5.2) and forms one of this study's main research contributions: An LLM-based value extraction method (see research contribution 1). As figure 3.4 illustrates, the value extraction module is realized as a *value extraction agent* that receives a set of value representations as input in addition to the pre-processed data set. The output of the module is a labeled data set, where the labels are personal values. This requires the value extraction module to implement a solution to sub-research question 3, as stated in section 1.5.2: *How to determine for a piece of text d* $\in D_c$ *and a context-specific value* $v \in V_c$*, whether v is referenced in d?*

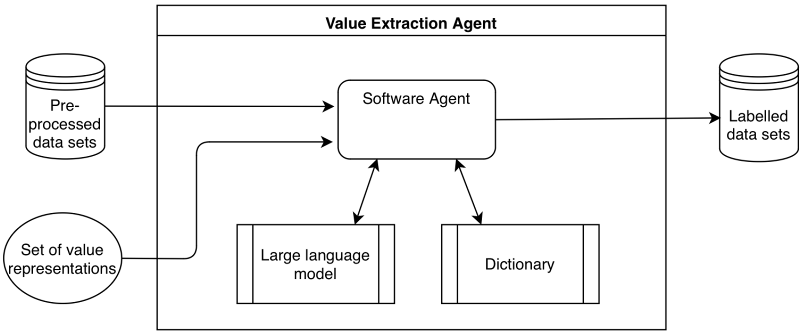


FIGURE 3.4: Section of the VIVE pipeline with the value extraction agent.

VIVE uses a combination of large language models (LLMs) and a dictionary for the task of value extraction. Algorithm 1 shows how, for given a natural language data set D_c, VIVE extracts for each text $d \in D_c$, the referenced values V_d.

Algorithm 1 VIVE Value Extraction

```
 1: Input: dataset $D_c$, task type $T$ (multi-label or single-label), combination strategy
    $strategy$, set of context-specific values $V_{c,A}$
 2: $D_{labeled} \leftarrow \emptyset$
 3: for $d \in D_c$ do
 4:     $V_d \leftarrow \emptyset$
 5:     if $T$ is multi-label then
 6:         $V_{d,llm} \leftarrow llm\_multi\_value\_extraction(d, V_{c,A})$
 7:         $V_{d,dict} \leftarrow dict\_multi\_value\_extraction(d, V_{c,A})$
 8:         $V_d \leftarrow combine\_multi\_value\_extraction\_sources(V_{d,llm}, V_{d,llm}, strategy)$
 9:     else
10:         $V_{d,llm} \leftarrow llm\_single\_value\_extraction(d, V_{c,A})$
11:         $V_{d,dict} \leftarrow dict\_single\_value\_extraction(d, V_{c,A})$
12:         $V_d \leftarrow combine\_single\_value\_extraction\_sources(V_{d,llm}, V_{d,llm})$
13:     end if
14:     $D_{\text{labeled}}$.add(<$d$, $V_d$>)
15: end for
16: Output: Labeled dataset $D_{\text{labeled}}$
```

First, the values extracted by the LLM $V_{d,llm}$ and the values extracted by the dictionary $V_{d,dict}$ are obtained separately (see lines 6-7 and 10-11 in algorithm 1). In a second step, $V_{d,llm}$ and $V_{d,dict}$ are combined to obtain the final set of referenced values V_d (see lines 8 and 12 in algorithm 1). VIVE is designed to support both multi-label and single-label value extraction, as specified in the definitions 6 and 7. The individual steps of algorithm 1 are explained in the following subsections. Subsection 3.3.2 describes the functions $llm\_multi\_value\_extraction(d, V_{c,A})$ and $llm\_single\_value\_extraction(d, V_{c,A})$, subsection 3.3.3 describes

$dict\_multi\_value\_extraction(d, V_{c,A})$ and $dict\_single\_value\_extraction(d, V_{c,A})$, and subsection 3.3.4 describes $combine\_multi\_value\_extraction\_sources(V_{d,llm}, V_{d,llm})$ and $combine\_single\_value\_extraction\_sources(V_{d,llm}, V_{d,llm})$.

3.3.1 Pre-Processing

Preceding algorithm 1 are any pre-processing steps that are applied to the data. All pre-processing steps should aim to minimize the risk that the data is manipulated in a way that changes what underlying values are extracted. To ensure that personal values are extracted accurately, pre-processing should involve as minimal alteration of a given text as possible. Context-aware pre-processing techniques, such as lemmatization, are preferred. We adopt the definition of context-aware techniques provided by Saleem et al. [62]. Context-aware techniques are methods that preserve the semantic meaning and nuances of a text by considering the context of each word.

3.3.2 Extraction with Large Language Models

Large language models (LLMs) can extract personal values from natural language text via inference. For background on the functionality of LLMs and an explanation of the LLM inference function, see section 2.4.3. Extracting values via LLM inferences requires developing a prompt strategy and determining the extracted values from the LLM inference output. Addressing sub-research question 3, from section 1.5.2, LLMs provide the following answer: A context-specific value $v \in V_c$ is referenced in a natural language text d, if the inference output $LLM_{inference}(p)$ for a prompt $p = prompt(d, v)$ is affirmative. In simple terms, a value v is referenced in a text d if the answer to the question is *"Is value v referenced in text d?"*, given by the LLM, is affirmative.

$$f : d \in D_c, v \in V_c \mapsto \begin{cases} 1 & ; \quad LLM_{inference}(d, v) is\ affirmative \\ 0 & ; \quad LLM_{inference}(d, v) is\ negative \end{cases} \tag{3.1}$$

The way an LLM is prompted generally directly influences the inference output. We use the term *prompt strategy* to refer to the way we formulate prompts. We differentiate between the *multi-label* and the *single-label* prompt strategy. For a given text $d \in D_c$ the multi-label prompt strategy is applied to extract the set of referenced values V_d, whereas the single-label prompt strategy is applied to extract the primary referenced value v_d. This differentiation corresponds with the definitions from section 1.5.1, for value extraction as a multi-label task (definition 6) and as a single-label task (definition 7). The below sections describe in detail the functionality of the VIVE value extraction module for the multi-label and single-label value extraction respectively. In principle, various prompt strategies are conceivable for the task of value extraction. However, a comprehensive evaluation of different prompt strategies exceeds the scope of this study. Section 6.2.3 describes the exploration of different prompt strategies as a possible direction of future work.

LLM Multi-label Value Extraction

Algorithm 2 illustrates the multi-label value extraction. For a given text d and set of identified context-specific values $V_{c,A}$, VIVE uses the *MULTI_LABEL_PROMPT_TEMPLATE* to prompt the LLM once for each personal value v in $V_{c,A}$.

MULTI_LABEL_PROMPT_TEMPLATE(d, v):
"Is the following personal value an underlying value for the following text?
Text: <TEXT>
Personal Value: <VALUE NAME>
To answer the question, consider, whether the following sentence is a correct statement:
The author composed this text, because <VALUE NAME> is important to him/her?
Answer only with 'yes' or 'no'!"

The placeholder <*TEXT*> is replaced with a given text d and the placeholder <*VALUE NAME*> is replaced with the name of a context-specific value. Provided the prompt template, the prompt strategy to extract the set of referenced context-specific values V_d for a text d works as follows: For each personal value v in the set of identified context-specific values $V_{c,A}$, the LLM is prompted with the prompt-template, replacing <*TEXT*> with text d and <*VALUE NAME*> with *v.name*, the name of the value v. As the last sentence in the prompt template indicates (*"Answer only with 'yes' or 'no'!"*), the LLM inference output is expected to be either "yes" or "no". If the inference output is "yes", we consider the value v as referenced in the text d. If the inference output is "no", we consider the value v as not referenced in the text d.[1] Finally, the set of referenced context-specific values V_d for the text d contains all values for which the inference output is "yes".

Algorithm 2 llm_multi_value_extraction

```
Input: natural language string $d$, set of context-specific values $V_{c,A}$
Output: set of referenced values $V_d$
$V_d \leftarrow \emptyset$
for $v \in V_{c,A}$ do
    $prompt \leftarrow$ MULTI_VALUE_PROMPT_TEMPLATE($d, v$)
    $inference\_output \leftarrow LLM_{inference}(prompt)$
    if $inference\_output$ == "Yes" then
        $V_d$.add($v$)
    end if
end for
return $V_d$
```

LLM Single-label Value Extraction

Algorithm 3 illustrates the single-label value extraction. For the single-label task, VIVE applies a different prompt strategy that consists of two consecutive prompts. Through *SINGLE_LABEL_PROMPT_TEMPLATE_1* the LLM is asked to indicate which one of the values in $V_{c,A}$ is the primarily referenced value for a given text d. The placeholder <*TEXT*> is replaced with a given text d and the placeholder <*VALUE NAME FOR EACH VALUE*> is replaced with a list of the names of all values from the set of context-specific values $V_{c,A}$, separated by a comma.

[1] In the case that the inference output is neither "yes" nor "no", the LLM is prompted again. When prompting the LLM again, the algorithm essentially starts a loop. Therefore, it is advised to set a maximum number of prompts per text d, to avoid infinite loops. The case that the LLM inference output is not as instructed in the prompt template can occur due to the elements of randomness present in LLMs (see section 2.4.3).

> **SINGLE_LABEL_PROMPT_TEMPLATE_1(d, $V_{c,A}$)**:
> *"Which of the following personal values is the underlying value of the following text?*
> *Text: <TEXT>*
> *Personal Values: <VALUE NAME FOR EACH VALUE>*
> *To answer the question, consider, which one of the personal values is the most important to the author of the text.*
> *Answer only by stating the underlying values in this format [<value>]! Do not give any additional information."*

We consider the value $v \in V_{c,A}$, whose name is present in the inference output, as the extracted value v_d of text d.[2] However, the way we formulate the *SINGLE_LABEL_PROMPT_TEMPLATE_1*, encourages the LLM to always pick one value. To allow for the option that a text does not reference any of the values in $V_{c,A}$, we "double-check" the indicated value. We do this by prompting the LLM a second time with the *SINGLE_LABEL_PROMPT_TEMPLATE_2*.

> **SINGLE_LABEL_PROMPT_TEMPLATE_2(d, v)**:
> *"Is the following personal value an underlying value for the following text?*
> *Text: <TEXT>*
> *Personal Value: <VALUE NAME>*
> *To answer the question, consider, whether the following sentence is a correct statement: The author composed this text, because <VALUE NAME> is important to him/her?*
> *Answer only with 'yes' or 'no'!"*

Algorithm 3 llm_single_value_extraction

```
Input: natural language string $d$, set of context-specific values $V_{c,A}$
Output: set with one referenced value $V_d$
$V_d \leftarrow \emptyset$
$prompt \leftarrow$ SINGLE_VALUE_PROMPT_TEMPLATE_1($d, V_{c,A}$)
$inference\_output \leftarrow LLM_{inference}(prompt)$
for $v \in V_{c,A}$ do
    if $v.name \in inference\_output$ then                ▷ If $v.name$ is a substring
        $prompt \leftarrow$ SINGLE_VALUE_PROMPT_TEMPLATE_2($d, v$)
        $inference\_output \leftarrow LLM_{inference}(prompt)$
        if $inference\_output$ == "Yes then
            return $[v]$
        end if
    end if
end for
return $\emptyset$
```

[2] As indicated by the last sentence of the prompt template, the LLM inference output is expected to contain only the name of one value and no additional information. In case no value name or more than one value name is present in the inference output, the LLM is prompted again. Similarly to the repeated prompting in the multi-label case (see section 3.3.2), it makes sense to set a maximum number of prompts per text d.

3.3.3 Extraction with a Dictionary

We define a dictionary as a lookup table with personal values as the keys and keyword lists as the values. Basically, for each context-specific personal value, the dictionary stores a list of keywords. Given a natural language text, the dictionary-based algorithm can extract personal values from it by checking if any of the stored keywords occur in the text. For each keyword that occurs, the dictionary-based algorithm determines that the associated value is referenced. Pre-processing steps, such as lemmatization, can increase the amount of values that can be extracted by a dictionary-based algorithm. For sub-research question 3, from section 1.5.2, the dictionary approach provides the following answer, where k is a keyword and *Dict[v]* is the list of keywords that are stored for value v in the dictionary.

$$f : d \in D_c, v \in V_c \mapsto \begin{cases} 1 & ; \quad \exists k \in Dict[v] : k\ occurs\ in\ d \\ 0 & ; \quad otherwise \end{cases} \tag{3.2}$$

Dictionary construction

The dictionary requires a set of initial keywords to be bootstrapped. These initial keywords are obtained during the value identification. If no keywords can be obtained during the value identification, an additional step might be necessary, for example, a manual annotation of keywords. Given the initial keywords, VIVE expands the dictionary to make it more powerful. We use the term *powerful* in this context as a function of how many values a dictionary extracts. The higher the number of extracted values for a given data set on average, the more powerful a dictionary is. Note that a more powerful dictionary does not necessarily increase the quality of the extraction. Generally, the more the dictionary is expanded, the more values it predicts on average for a given message. The "amount" of dictionary expansion therefore influences the precision and recall of the value extraction. More specifically, a more extensive expansion leads to higher recall and lower precision. The method chosen for this study aims to strike a balance as best as possible. Fundamentally, our reason for expanding the dictionary is the assumption that the few keywords that are manually annotated during the value identification do not yield a powerful enough dictionary. More specifically, we hypothesize that without dictionary expansion, only a small subset (less than 50 percent) of the personal values referenced in a text would be extracted.

Principally, many methods are conceivable for the dictionary expansion. VIVE makes use of two methods for the dictionary expansion - synonyms and word embedding similarities. When expanding the dictionary with synonyms, we simply consider all existing synonyms of the initially present keywords as keywords themselves. For example, if *housing* is annotated as a keyword for the personal value *shelter*, its synonym *accommodation* also becomes a keyword for *shelter*. When expanding the dictionary with a word embedding model, we consider for each initially present keyword the n terms that are closest in the embedding space.

Dictionary Multi-Label Value Extraction

Algorithm 4 shows how the dictionary extracts personal values. As mentioned above, for a given natural language string d, the dictionary checks for each value v in the set of context-specific values $V_{c,A}$, if any stored keywords appear in d.

Algorithm 4 dict_multi_value_extraction

```
Input: natural language string $d$, set of context-specific values $V_{c,A}$
Output: set of referenced values $V_d$
$V_d \leftarrow \emptyset$
for $v \in V_{c,A}$ do
    for $keyword \in K[v]$ do
        if $keyword \in d$ then                              ▷ If $keyword$ is a substring of d
            $V_d$.add($v$)
            break                   ▷ Stop checking further keywords for this value
        end if
    end for
end for
return $V_d$
```

Dictionary Single-Label Value Extraction

Algorithm 5 shows how the dictionary extracts the primarily referenced value v_d for a given natural language string d. This refers to single-label value extraction. As algorithm 5 shows, the primarily referenced value is extracted by first extracting the set of all referenced values V_d. In the case that V_d is empty or contains exactly one value, V_d is returned as it is. In the case the V_d contains more than one value, the algorithm randomly chooses a value $v \in V_d$.

Algorithm 5 dict_single_value_extraction

```
Input: natural language string $d$, set of context-specific values $V_{c,A}$
Output: set with one referenced value $V_d$
$V_d \leftarrow dict\_multi\_value\_extraction(d, V_{c,A})$
if $V_d = \emptyset$ or $|V_d| = 1$ then
    return $V_d$                                         ▷ Return the extracted value
else
    Randomly choose a value $v$ from $V_d$
    return $[v]$                                                  ▷ Choose randomly
end if
```

3.3.4 Combining Value Extraction Sources

VIVE is designed to use a combination of a dictionary and large language models to perform value extraction. Both, the dictionary and LLMs, represent generic value extraction sources. We define a value extraction source as a method that provides an answer to sub-research question 3: *What is a function that indicates for any combination of a piece of text $d \in D_c$ and a context-specific value $v \in V_c$, whether v is referenced in d?* Given a set of context-specific values V_c and a piece of text d, a value extraction source S extracts a set of values $V_{d,S} \subseteq V_c$. Because VIVE uses more than one value extraction source it requires a strategy to combine the extracted values of all its value extraction sources. As an example, we can consider the message d = *"Hello, is the program for free housing for Ukrainians still valid?"* from the Ukraine data set. The dictionary extracts V_{dict} = {shelter}, but the LLM extracts V_{LLM} = {shelter, staying warm in winter}.

Combining Value Extraction Sources for Multi-Label Value Extraction

In principle, various ways of combining the output of multiple value extraction sources are conceivable. For the multi-label task, two simple strategies are to take either the union or the intersection of the outputs of all available value extraction sources. Taking the union means the value extraction method optimizes for recall. In the above example, taking the union results in V_d = {shelter, staying warm in winter}. In contrast, taking the intersection means optimizing for precision, because a value is only extracted if it is extracted by each of the available value extraction sources. In the example, this results in V_d = {shelter}. The default strategy VIVE applies is to optimize for recall. Algorithm 6 illustrates the combination of value extraction sources for the multi-label task, given a text d and the set of outputs of all available value extraction sources, namely the LLM $V_{d,llm}$ and the dictionary $V_{d,dict}$

Algorithm 6 combine_multi_value_extraction_sources

```
**Input:** set of values extracted the LLM $V_{d,\text{llm}}$, set of values extracted by the Dictionary $V_{d,\text{dict}}$, combination strategy *strategy*
**Output:** combined set of referenced values $V_d$
$V_d \leftarrow \emptyset$
**if** $strategy = optimizing\_for\_recall$ **then**
    $V_d \leftarrow V_{d,\text{llm}} \cup V_{d,\text{dict}}$        ▷ Union of extracted values
**else if** $strategy = optimizing\_for\_precision$ **then**
    $V_d \leftarrow V_{d,\text{llm}} \cap V_{d,\text{dict}}$        ▷ Intersection of extracted values
**end if**
**return** $V_d$
```

Combining Value Extraction Sources for Single-Label Value Extraction

Algorithm 7 illustrates the combination of value extraction sources for the single-label task, given a text d, the set of values extracted by the dictionary $V_{d,dict}$, and the set of values extracted by the LLM $V_{d,LLM}$. Single-label value extraction implies that only one or no value should be extracted for a given text d. Therefore, taking the union or intersection of the outputs of all available value extraction sources is not a viable option. Instead, VIVE combines the single-label outputs of the LLM and the dictionary as follows: If the LLM and the dictionary extract the same value, we consider that value as the referenced value v_d of text d (see algorithm 7, lines 3 to 5). If either the LLM or the dictionary extracts a value and the other value extraction source does not extract a value, we consider the extracted value as the referenced value v_d, again optimizing for recall (see algorithm 7, lines 6 to 8 and lines 9 to 11). If neither value extraction source extracts a value, we assume text d to not reference any of the context-specific values (see algorithm 7, lines 12 to 13). Finally, if the LLM and the dictionary extract two different values, the algorithm randomly chooses one of the two values (see algorithm 7, lines 14 to 17).

Algorithm 7 combine_single_value_extraction_sources

```
**Input:** $V_{d,\text{llm}}$, $V_{d,\text{dict}}$
**Output:** combined referenced value $V_d$
**if** $V_{d,\text{dict}} \neq \emptyset$ & $\mathrm{V}_{d,\text{LLM}} \neq \emptyset$ & $\mathrm{V}_{d,\text{dict}} = V_{d,\text{LLM}}$ **then**
    $v_d \leftarrow V_{d,\text{dict}}$                              ▷ Both extract the same value
    **return** $[v_d]$
**else if** $V_{d,\text{dict}} \neq \emptyset$ & $\mathrm{V}_{d,\text{LLM}} = \emptyset$ **then**
    $v_d \leftarrow V_{d,\text{dict}}$                              ▷ Only dictionary extracts a value
    **return** $[v_d]$
**else if** $V_{d,\text{dict}} = \emptyset$ & $\mathrm{V}_{d,\text{LLM}} \neq \emptyset$ **then**
    $v_d \leftarrow V_{d,\text{LLM}}$                               ▷ Only LLM extracts a value
    **return** $[v_d]$
**else if** $V_{d,\text{dict}} = \emptyset$ & $\mathrm{V}_{d,\text{LLM}} = \emptyset$ **then**
    **return** $\emptyset$                                          ▷ Neither extracts a value
**else if** $V_{d,\text{dict}} \neq V_{d,\text{LLM}}$ **then**
    $V_d = \leftarrow V_{d,\text{llm}} \cup V_{d,\text{dict}}$
    Randomly choose a value $v_d$ from $V_d$
    **return** $[v_d]$                       ▷ LLM and dictionary extract different values
**end if**
```

Chapter 4

Instantiation of VIVE to the Red Cross Case Study

Section 3 gives a conceptual overview of VIVE. This section describes how we apply the VIVE pipeline to the requirements of the Netherlands Red Cross regarding the processing of feedback data from humanitarian programs (see section 1.4.1). Figure 4.1 presents an overview of how we instantiate the VIVE pipeline and the subsequent sections describe the realization of the individual modules of the pipeline in detail.

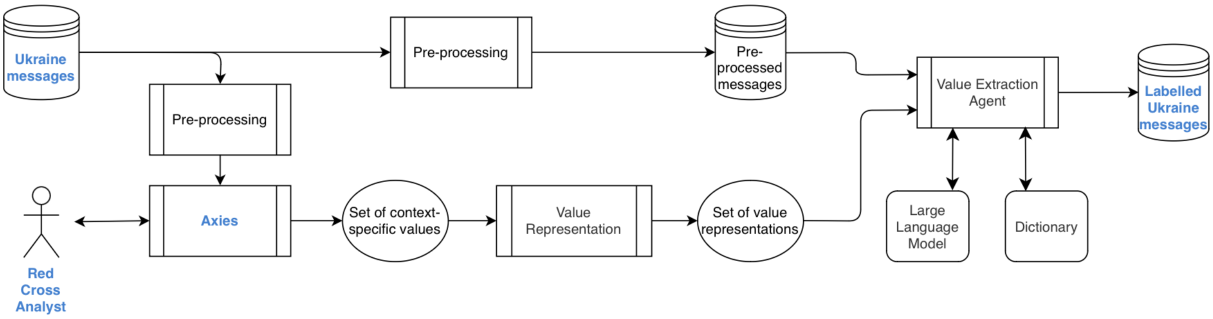


FIGURE 4.1: VIVE pipeline for the context of the Ukraine data set

4.1 Data set

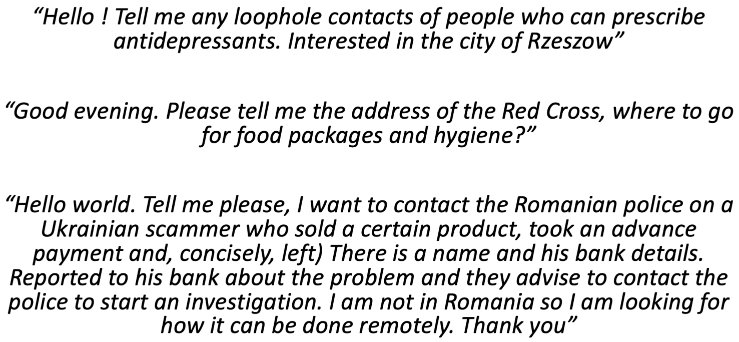
As mentioned in the introduction (see section 1.4.1), the Netherlands Red Cross collects chat messages from public Telegram groups, that are used by Ukrainian refugees or internally displaced people[1]. While a Telegram group is not by design constrained to a location, most such groups pertain to a specific region that is usually indicated by the name of the group. Some example group names are *kishineww*, *moldovaukraina*, or *help_ukraina*. A complete list of all groups from which messages are collected can be found in Appendix A. The Netherlands Red Cross collects messages from Telegram groups from Bulgaria, Hungary, Montenegro, Poland, Slovakia, and Ukraine. For each message, the following information is collected:

- id_post: Chat-specific ID of the message.
- country: Country in which the group is used, for example *"MDA"*.

[1] Information about the project can be found under:
`https://510.global/product/sml/`,
`https://github.com/rodekruis/social-media-listening/tree/master`

- source: Public link to the Telegram group, for example *"t.me/kishineww"*.
- datetime_scraped: Date and time the message was scraped from the group, for example, *"2023-10-02 06:21:08.933"*.
- datetime: Date and time the message was posted, for example *"2023-11-30 18:32:01"*.
- date: Date the message was posted, for example *"2023-04-09"*.
- text_post: Actual contents of the message.

For this study, the Netherlands Red Cross provides a subset of the collected data. While the original messages are in Ukrainian, the provided subset of messages is translated to English. The messages are anonymized, by replacing names, phone numbers, URLs, and other identifiers with a placeholder. For example, <*PERSON*> serves as a placeholder for names. In the following this subset of the collected Telegram messages is referred to as the *Ukraine data set* and constitutes the context for this instantiation of VIVE, following definition 2 of the term *context*. The Ukraine data set contains a total of 4522 messages. Figure 4.2 shows three example messages from the Ukraine data set.



"Hello ! Tell me any loophole contacts of people who can prescribe antidepressants. Interested in the city of Rzeszow"

"Good evening. Please tell me the address of the Red Cross, where to go for food packages and hygiene?"

"Hello world. Tell me please, I want to contact the Romanian police on a Ukrainian scammer who sold a certain product, took an advance payment and, concisely, left) There is a name and his bank details. Reported to his bank about the problem and they advise to contact the police to start an investigation. I am not in Romania so I am looking for how it can be done remotely. Thank you"

FIGURE 4.2: Example messages from the Ukraine data set.

4.2 Value Identification

For this study, we use a value identification method based on *Axies*. Axies is a methodology to identify context-specific values, proposed by Liscio et al. [39]. This section motivates the chosen method and explains the modifications that were done.

4.2.1 Pre-processing

As a pre-processing step, we filter the Ukraine data set to remove as many messages as possible that cannot be used for value identification, for example, spam messages. Table 4.1 shows how many messages were removed through each of the criteria. Firstly, all messages with more than 500 characters or less than 50 characters are removed. This decision is based on the fact that most spam messages contain more than 500 characters and many short messages simply contain a confirmation, for example *"Okay, I understand!"*. Overall, the filtering of the Ukraine data set focuses on keeping only messages that have a high chance of allowing the identification of underlying values due to the limited annotation possibilities for this study, which section 5.2.1 elaborates on.

TABLE 4.1: Number of removed messages

criteria	total messages removed	% of messages removed
max. length	276	6.1
min. length	285	6.3

4.2.2 Axies

Axies is a hybrid intelligence method that enables the identification of a set of context-specific personal values through an annotation process [39]. Section 2.3.1 provides background on the Axies methodology and figure 2.2 shows an overview of it. For this study, we choose Axies over other available value identification methods (for examples see [10][76]) for the following reasons: 1. Axies provides a hybrid intelligence solution, 2. Axies is built specifically for data sets of short natural language texts, 3. Axies is designed to output context-specific personal values, 4. Axies uses a value representation that is suitable for a subsequent extraction of values via a large language model. These features align well with the elicited requirements from section 1.4.

For this study, we applied some minor conceptual modifications to the original version of Axies. The following describes the modifications and their motivations.

Modification 1 (Multiple values per message)**.** *Given a context, the original version of Axies focuses solely on the creation of a context-specific set of personal values and does not concern the task of value extraction or the question of where in the data certain values are referenced. In contrast, this study uses Axies as a necessary prerequisite for value extraction. Therefore we aimed to collect as much information about references of personal values, without significantly increasing the effort for the annotator. We modified Axies so that for a shown message, the annotator has the option to both, annotate new values or select already existing values.*

Modification 2 (Potential related messages)**.** *For this study, the exploration of potentially related messages is added as an additional step to the original version of the Axies exploration workflow (see "Annotate potential related messages" in figure 4.3). We modified Axies so that whenever a new personal value is annotated, the annotator is shown messages that are semantically similar to the message for which the new value was annotated. The reasoning behind this is that semantically similar messages likely reference the same personal values.*

Modification 3 (Single annotator consolidation)**.** *The original version of Axies includes a consolidation phase in which the individually derived value lists of all annotators are merged into one final list. In this study, the consolidation phase of the annotation is done by only one analyst from the Netherlands Red Cross. Therefore, we simplify the consolidation phase to the following steps to be performed by the annotator: 1. Deleting values from the list (optional), 2. adding or deleting keywords for values (optional), 3. Add a description for each value (required).*

Figure 4.3 shows the adapted workflow that the annotator goes through during the exploration phase. The annotator gets prompted with one message at a time. For a given message, the annotator has to decide whether any personal values are referenced. If not, the annotator indicates whether this is because the message is not comprehensible or simply no value is referenced. If the annotator decides that a personal value is referenced, there are two possible cases: The value has already appeared in a previously annotated message or it appears for the first time. In the latter case, the annotator is asked to give the new value a name and add it to the list of identified values. In both cases, keywords can be added to the annotated value. The annotator can decide at any moment that the annotation is complete.

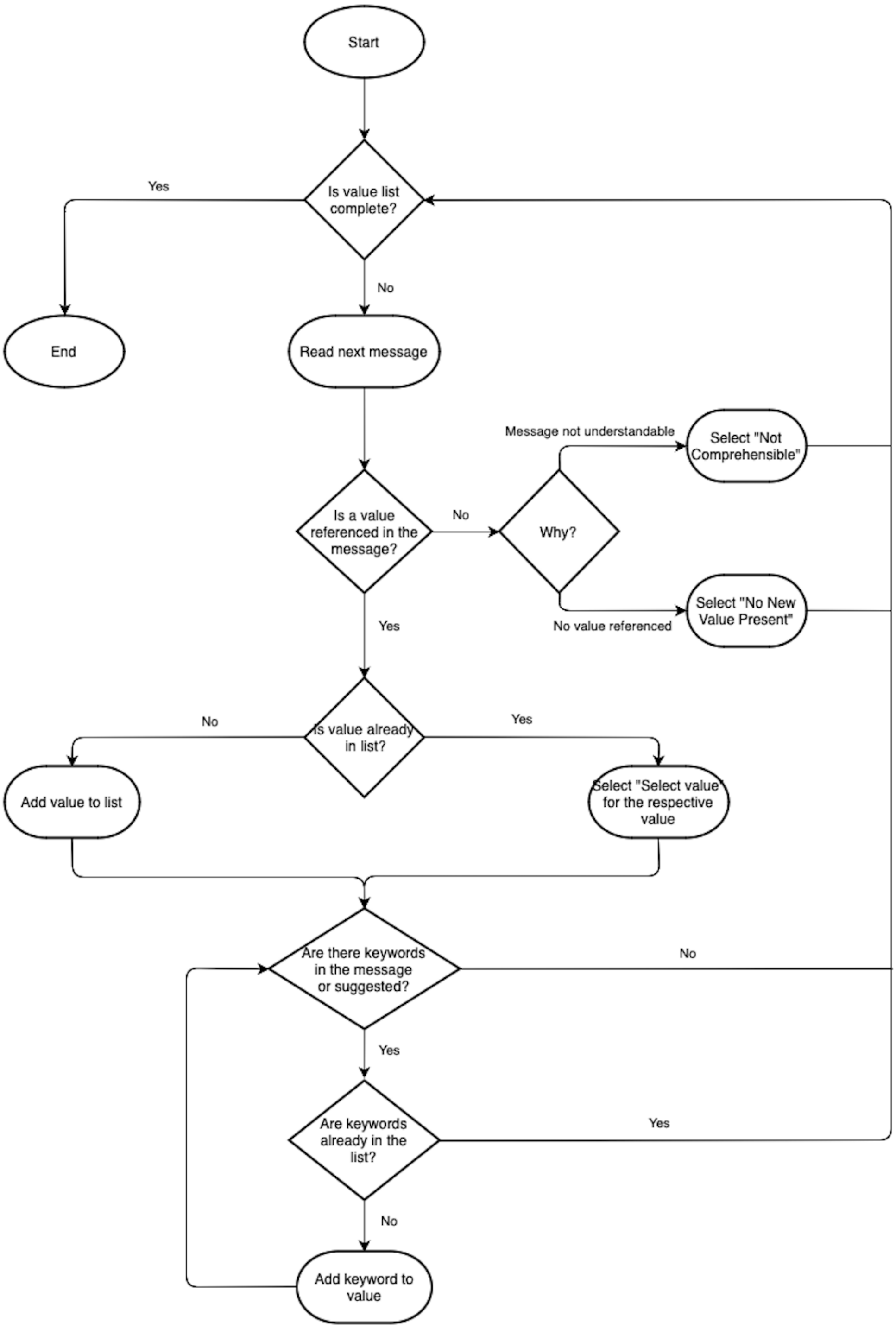

Start
Is value list complete?
Yes
No
End
Read next message
Message not understandable
Select "Not Comprehensible"
Is a value referenced in the message?
No
Why?
Select "No New Value Present"
No value referenced
Yes
Is value already in list?
No
Yes
Add value to list
Select "Select value" for the respective value
Are there keywords in the message or suggested?
No
Yes
Are keywords already in the list?
Yes
No
Add keyword to value

FIGURE 4.3: Adapted Axies workflow of the user during the exploration phase

4.3 Value Representation

As described in section 3.2, VIVE represents personal values as triples <*n, K, D*>, with a name *n*, a list of keywords *K*, and a description *D*. In this study, the value representation format after the value identification is given by Axies. The Axies output value representation contains all elements of the VIVE value representation, therefore the translation between the two representations becomes trivial. Table 4.2 shows some example VIVE value representations of personal values that were identified for the Ukraine data set. A table including all identified personal values for the Ukraine data set can be found in Appendix B.

TABLE 4.2: Example values from Ukraine context in Axies value representation

Name	**Keywords**	**Description**
shelter	housing	Ukrainian refugees need a shelter/place to stay or to live when they flee to other regions of Ukraine or to other/neighbouring countries. This is of paramount importance.
mental health	psychological support, psychological health	They look for and offer psychological support because many people are traumatized by war.
staying connected	mobile phone, connectivity, Internet	Good and inexpensive service provider (mobile phone, Internet) is very important to stay in touch with their families/husbands.
disappointment in this city/country	go back home	From time to time we see messages when people get disappointed in a new place and decide to go back to Ukraine even though it is unsafe over there.
help for refugees	humanitarian aid, Red Cross	Humanitarian and other kinds of help is needed for the people affected.

4.4 Value Extraction

As mentioned in section 3.3, VIVE uses a combination of large language models (LLMs) and a dictionary for the task of value extraction. The following describes how the value extraction module is instantiated for the Ukraine data set. Section 4.4.1 describes in detail how we extract values with LLMs, relating to section 3.3.2 and section 4.4.2 describes how we extract values with a dictionary, relating to section 3.3.3.

4.4.1 Extraction with Large Language Models

For this study, we utilize quantized LLMs (QLLMs). By using lower precision numbers to represent the model's weights, QLLMs significantly reduce the computational demands of LLM inference. We utilize QLLMs firstly, for sustainability reasons, to reduce energy consumption during inference and secondly, due to the limited computational resources available for this study. Recent studies show that quantization techniques can effectively reduce the computational resources that are required for inference while maintaining model performance [37][79].

Prompt strategy

As explained in section 3.3.2, VIVE uses LLM inference to extract personal values. VIVE provides prompt templates for multi-label and single-label value extraction. In this instantiation of VIVE, we make a slight modification to the prompt templates, by replacing all occurrences of the term *text* with the term *message*. This is because the texts in the Ukraine data set are exclusively messages, written by people on Telegram. Therefore, we consider the term *message* as slightly more precise in this context. In addition to a prompt, the LLM receives context information. Generally, the context information is provided before prompting and includes any context information or specific instructions needed for the LLM to properly understand the prompt. For this study, we use the following two contexts:

> **Context 1: Full value representation**
> *"You are an analyst from a humanitarian organization. You have collected messages from social media groups, in which war refugees ask questions and get information. You have identified the following personal values that people deem particularly important in the context of fleeing the war in their country and settling in a different country:*
> *FOR EACH VALUE:*
> *Personal Value 1: <VALUE NAME>*
> *Keywords: <VALUE KEYWORDS"*
> *Description: <VALUE DESCRIPTION>"*
>
> **Context 2: Simple value representation**
> *"You are an analyst from a humanitarian organization. You have collected messages from social media groups, in which war refugees ask questions and get information. You are interested in the personal values that someone references."*

The main difference between contexts 1 and 2 is the value representation. In context 1 the placeholder "FOR EACH VALUE" indicates that for each value v in the set of context-specific values $V_{c,A}$, the name of the value (= <VALUE NAME>), any given keywords (= <VALUE KEYWORDS>), and the description (= <VALUE DESCRIPTION>) are listed. Hence, it contains full representations as described in section 3.2. In contrast, context 2 does not contain any information about the values.

4.4.2 Extraction with a Dictionary

In this instantiation of VIVE, the initially present keywords stem from the annotator that performed the value identification via Axies. Given these manually annotated keywords, we expand the dictionary as described in 3.3.3. For the expansion through word embedding similarities, we set the parameter n to 5. This means, that for each of the initially present keywords, the five most similar words in the embedding model are added as keywords to the dictionary. As a pre-processing step, we use lemmatization to make a keyword match more likely and hence, make the dictionary more powerful.

Chapter 5

Evaluation

This section describes how we evaluate VIVE, the experiments that we perform through our instantiation of VIVE, and presents the results we obtain. The entire implementation can be found in this project's GitHub repository [1]. The experiments conducted for this study aim to test the proposed method - VIVE - under consideration of the main research question and sub-research questions 3 and 4 (see section 1.5.2). Furthermore, it aims to answer the experimental research questions 1, 2, and 3. Section 5.1 describes the method we follow to answer the experimental research questions. Section 5.2 describes our experimental setup, including how we conduct the annotation of the data (5.2.1), how we define the evaluation metrics (5.2.2), and a summary of the implementation (5.2.3). Section 5.3 reports all results, based on which section 5.4 discusses experimental research question 1, section 5.5 discusses experimental research question 2, and section 5.6 discusses experimental research question 3.

As mentioned in section 1.2, in the introduction, we hypothesize that in principle LLMs possess sufficient natural language understanding capabilities to extract personal values from natural language text. However, for many tasks, LLMs tend to lack precision. On the other hand, dictionaries tend to lack recall but can be very precise, especially when using keywords that were annotated by a domain expert. We hypothesize that LLMs and dictionaries possess complementary strengths when it comes to extracting personal values from text.

Experimental Research Question 1. *Does the use of a combination of LLMs and a dictionary increase the accuracy of the value extraction, compared to using either LLMs or a dictionary alone?*

Hypothesis 1. *The combination of LLMs and a dictionary for value extraction leads to significantly higher accuracy compared to using either LLMs or a dictionary alone.*

Large language models are an integral part of the value extraction module of VIVE. Therefore, for the evaluation of the VIVE pipeline, we deem it relevant to examine the impact of the choice of LLMs on the value extraction. Note that the experimental research question 2 and corresponding hypothesis 2 presuppose the use of state-of-the-art LLMs. Meaning, that it is not of interest to examine whether a difference in performance can be observed within a selection of LLMs, where some models are much older than others. We utilize and evaluate the following three quantized large language models (QLLMs) for this study. This selection is largely based on the state-of-the-art performance of QLLMs at the time of writing. All three utilized model types are shown to have very little loss in performance through quantization [37].

[1] `https://github.com/brigoraoul/VIVE`

- **Llama3:** At the time of writing, Llama3 is the latest generation of the open-source Llama model family. In contrast to previous generations of the Llama model, Llama3 is trained on a larger dataset, consisting of over 15 trillion tokens collected from publicly available sources. Furthermore, it features improved token efficiency and advanced safety tools like Llama Guard 2 and Code Shield.

- **Mistral:** The Mistral 7B model [33] is optimized for efficiency and speed. It uses grouped-query attention (GQA), to accelerate the speed of inferences, and sliding window attention (SWA) to handle sequences of arbitrary length.

- **Gemma:** Gemma is a family of light-weight models, released by the Google DeepMind team [69]. In this study, we utilize the Gemma 7B model, which is pre-trained on 6 trillion tokens. It incorporates multi-query attention (MQA) to accelerate the speed of inferences.

Experimental Research Question 2. *Does the use of different large language models impact the accuracy of value extraction?*

Hypothesis 2. *The use of different state-of-the-art large language models for value extraction does not lead to significantly different accuracies.*

The basis for experimental research question 3 is our assumption that different value representations can significantly impact the accuracy and effectiveness of the value extraction. Based on this hypothesis, the value representation module is an integral part of the VIVE pipeline.

Experimental Research Question 3. *What is the impact of different value representations on the performance of value extraction?*

Hypothesis 3. *The accuracy of an LLM-based value extraction method is significantly higher when given a more extensive representation of values.*

5.1 Method

This section describes the method we follow, to answer the above experimental research questions 1, 2, and 3, broken down into the individual steps.

1. We perform value identification for the context of the Ukraine data set (see section 4.1) to obtain a set of context-specific values, utilizing Axies (see sections 2.3.1 and 4.2.2).

2. We prepare the identified context-specific values for value extraction by transforming them into the chosen value representation (see section 4.3).

3. We ask an analyst of the Netherlands Red Cross to select a subset of five personal values from the set of identified context-specific values for evaluation.

4. We select a subset of the Ukraine data set to be annotated with the selected subset of context-specific values.

5. The selected subset of the data is annotated by analysts from the Netherlands Red Cross. The data annotation process is described in detail in section 5.2.1.

6. We perform value extraction, as described in section 4.4, for the selected subset of the data and the selected subset of the context-specific values.

7. To assess experimental research question 1, we conduct experiment 1, in which we evaluate and compare three value extraction classifiers. These are a classifier that only uses a dictionary (*Dict*), a classifier that only uses a large language model (*llama3)*, and a classifier that uses a combination of the dictionary and a language model (*Dict+llama3)*.

8. To assess experimental research question 2, we conduct experiment 2, in which we take the best-performing classifier from experiment 1 (*llama3*) and evaluate and compare it against two more LLM-based value extraction classifiers (*mistral*, and *gemma*), using the three LLMs described above.

9. To assess experimental research question 3, we conduct experiment 3, in which we utilize the best-performing classifier from experiment 2 and evaluate and compare two versions of it (*llama3* and *Simple repr.*) that use different value representations.

10. To assess the usefulness of VIVE, we conduct a user study with analysts from the Netherlands Red Cross. Our user study design and the results are described in section 5.7.

Table 5.1 gives an overview of all six value extraction classifiers that we evaluate. The names that are assigned to the classifiers are also used in section 5.3 which presents the results of our evaluation. For each classifier, the listed value extraction sources refer to the methods that the classifier uses to extract values. We use the term *value extraction source* here as described in section 3.3.4. The column "LLM" indicates what, if any, LLM is used by the classifier. The names refer to the three LLM families described above. The column "Value Representation" indicates what value representation the classifier uses. The *full* value representation refers to the value representation described in section 3.2. The *simple* value representation only uses a name to represent values. For example, the personal value *help for refugees* in the *simple* value representation is represented only by its name ("help for refugees"), but not by any keywords or a description. On an implementational level, the difference between the *full* and the *simple* value representation lies in the contexts they use. The *full* value representation uses context 1 from section 4.4.1, whereas the *simple* value representation uses context 2.

Table 5.2 shows which value extraction classifiers are used in which experiments.

TABLE 5.1: Collection of value extraction classifiers used for the experimental evaluation.

Name	**Value Extraction Sources**	**LLM**	**Value Representation**
Dict	Dictionary	-	full
llama3	LLM	llama3	full
Dict+llama3	Dictionary, LLM	llama3	full
mistral	LLM	mistral	full
gemma	LLM	gemma	full
Simple repr.	LLM	llama3	simple

TABLE 5.2: Utilized value extraction classifiers per experiment.

Experiment	Value Extraction Classifiers
1	Dict, llama3, Dict+llama3
2	llama3, mistral, gemma
3	llama3, Simple repr.

We evaluate each classifier for single-label and multi-label value extraction by calculating accuracy, precision, recall, and the f1-score. Section 5.2.2 explains how we calculate these evaluation metrics. We compare the performance of classifiers by comparing their obtained accuracies via statistical tests. For experiments that include more than two classifiers, we perform an omnibus test to assess whether there is a significant difference between them. If the omnibus test indicates that there is a difference, we perform a pairwise post-hoc test. For experiment 3 which includes two classifiers, we perform a suitable statistical test for pairwise comparison. We thereby follow a test method demonstrated by Dell'Anna et al. [15].

5.2 Setup

This section details the experimental setup for the experiments we conduct to answer the experimental research questions 1, 2, and 3. The experimental setup includes the data annotation, in subsection 5.2.1, the used evaluation metrics, in subsection 5.2.2, and the implementation of VIVE, in subsection 5.2.3.

5.2.1 Data Annotation

For evaluation purposes, a subset of the Ukraine data set is annotated by analysts from the Red Cross. We sample a total of 170 messages to be annotated, using purposive sampling. With purposive sampling, samples are selected that are believed to be most relevant to the research objectives [28][77]. We choose this approach over random sampling because the labels that we define for annotation are sparse in the data and the annotation budget for this study is limited. More specifically, we aim to keep the time effort of each annotator to a maximum of one hour.

The annotation task is to decide for the selected messages which of the context-specific personal values that are identified during the value identification (4.2) are referenced. Due to the limited annotation budget, we select five of the identified values that were indicated as particularly interesting by a Red Cross analyst. The names of these five values are *shelter*, *mental health*, *staying connected*, *disappointment in this city/country*, and *help for refugees*. The full value representations of these five personal values are shown in table 4.2. Note that table 4.2 shows the value representations after value identification. It does not include any keywords that are added during the dictionary expansion, which is part of the value extraction (see section 3.3.3).

The annotation is performed in accordance with annotation guidelines, as summarized by Abualhaija et al. [1]. A message can reference one, more than one, or no values. Consequently, the annotation task is to determine, for each message, whether each of the five values is referenced. Two annotators label all 170 sample messages independently. We consider both annotators domain experts, as they are actively working on the SML project as employees of the Netherlands Red Cross.

It has to be noted that at the time of the annotation, we were in direct contact with the annotators and they were informed about the purpose of this study. To measure intercoder reliability we use the Cohen Kappa, because it is specifically designed for cases with two annotators [42]. The Cohen Kappa (CK) takes into account the observed agreement amongst raters p_o and the possibility of an agreement occurring by chance p_e and can therefore be viewed as a more refined measure than the percentage of agreement.

$$CK = \frac{p_o - p_e}{1 - p_e} \tag{5.1}$$

In this study, annotation conflicts are resolved by a third, independent expert from the Netherlands Red Cross. Out of the 170 annotated messages, the two annotators disagree on 24 messages. These 24 messages are resolved by the third, independent expert, by choosing one of the two annotations that is accepted as the correct one. The calculated Cohen's Kappa score for the two annotations is CK = 0.8459.

Table 5.3 shows for each of the five personal values, how many of the 170 messages are labeled with that value, after the resolution. Since we use the annotated data set as a test set for our evaluation, this distribution represents the distribution of labels in the test set. A total of 40 messages are annotated without any values ("no label") and a total of 3 messages are annotated with more than one value. The great majority of the messages are annotated with exactly one value.

TABLE 5.3: Value label distribution

Personal Value	**Number of annotations**
shelter	29
mental health	28
staying connected	29
disappointment in this city/country	12
help for refugees	34
no label	40

5.2.2 Evaluation Metrics

Principally, the task of value extraction is a multi-label task. Given a set of context-specific personal values $V_{c,A}$ and a piece of natural language text d, the task is to extract the set of referenced values $V_d \subset V_{c,A}$ (see definition 6). As definition 7 shows, value extraction can also be defined as a single-label task. In this case, the task is to extract the primarily referenced value $v \in V_{c,A}$ from d. Meaning the value that is most strongly referenced in d. As the algorithms in section 3 show, VIVE supports both multi-label and single-label value extraction.

For both cases, accuracy, precision, and recall have to be calculated differently. For the multi-label task, different ways of calculating accuracy, precision, and recall can be found in the literature [66]. In this study, we define accuracy, precision, and recall for the multi-label task as follows, based on definitions from Godbole et al.[25]. All measures are computed over n messages.

Accuracy The first accuracy measure for a multi-label task is the *exact match ratio* (MR). In the context of this study, an exact match is a message d from the data set, for which the exact, correct context-specific values are extracted, that is $V_d^{extracted} = V_d^{true}$

$$MR = \frac{1}{n} \sum_{d=1}^{n} I(V_d^{extracted} = V_d^{true}) \tag{5.2}$$

We define *partial correctness accuracy* (PC) as a second measure for accuracy to account for partial correctness of an extracted set of values V_d. PC defines the accuracy for each individual value extraction as the proportion of the correctly extracted values to the total number of values in $V_{c,A}$.

$$PC = \frac{1}{n} \sum_{d=1}^{n} \frac{\left|V_d^{extracted} \cap V_d^{true}\right|}{\left|V_d^{extracted} \cup V_d^{true}\right|} \tag{5.3}$$

Precision We define precision (P) as the proportion of predicted correct values to the total number of actual values.

$$P = \frac{1}{n} \sum_{d=1}^{n} \frac{\left|V_d^{extracted} \cap V_d^{true}\right|}{\left|V_d^{true}\right|} \tag{5.4}$$

Recall We define recall (R) as the proportion of correctly predicted values to the total number of predicted values.

$$R = \frac{1}{n} \sum_{d=1}^{n} \frac{\left|V_d^{extracted} \cap V_d^{true}\right|}{\left|V_d^{extracted}\right|} \tag{5.5}$$

F1-score In contrast to the accuracy, the F1-score takes into account the types of error, namely false positives and false negatives, that a classifier makes. It is particularly useful in this study because, as table 5.3 shows, the annotated data set used as the test set does not contain an equal number of samples for each class. We calculate the F1-score as a function of precision and recall.

$$F1 = 2 * \frac{P * R}{P + R} \tag{5.6}$$

For the single-label task, accuracy, precision, and recall are calculated according to their standard definitions. Furthermore, having only one label allows the calculation of confusion matrices. To allow a more in-depth analysis of the extraction of individual personal values, we calculate the precision, recall, and F1 per personal value. We do not calculate the accuracy per value because we do not have ground truth data for true negatives. Accuracy is commonly defined as the ratio of the total number of correctly predicted instances to the total number of instances. The total number of correctly predicted instances is the sum of correctly as true predicted and correctly as false predicted instances. Leaving out true negatives would result in the ratio of the true positives to the total number of instances, which does not correctly represent the performance of a classifier.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Number\ of\ Instances} \tag{5.7}$$

5.2.3 Implementation

This section briefly describes the implementation of the value identification, value representation, and value extraction modules, that are utilized for the instantiation of the VIVE pipeline to the Ukraine data set (see section 4).

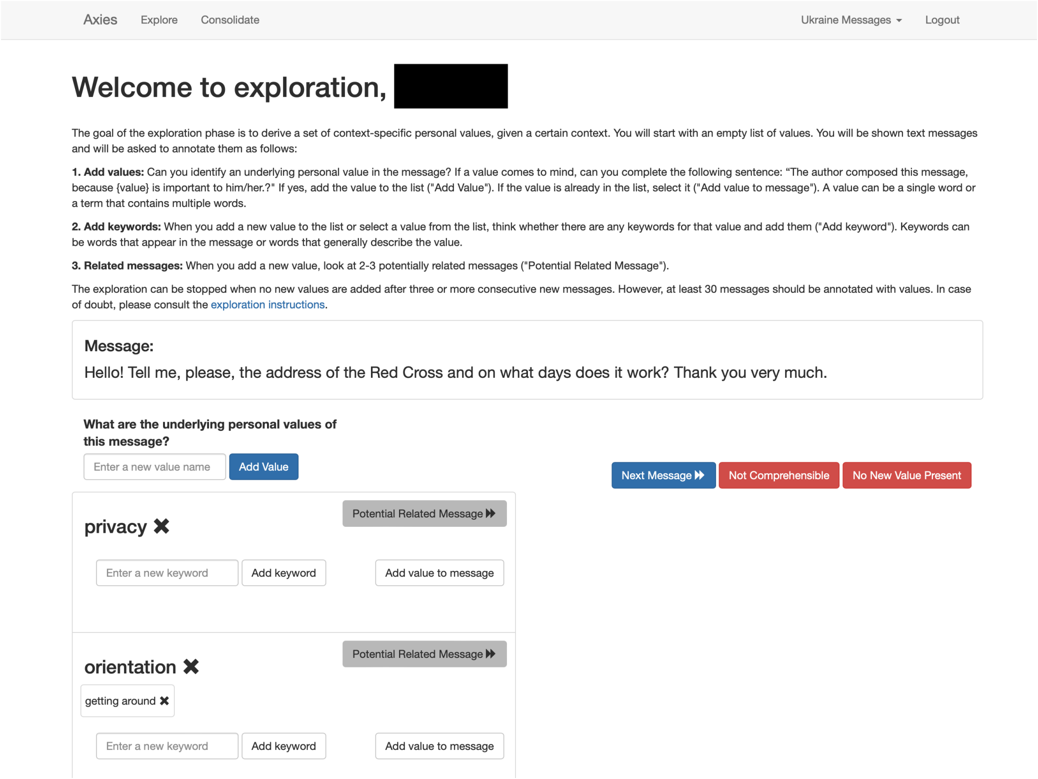
Value Identification

As explained in section 4.2, in this experiment we identify context-specific personal values through the Axies methodology. For this, we make use of a web platform built by Liscio et al. [38] that provides a tool for a guided value annotation of the data. An annotator can access the web platform online and create an account. The web platform guides the annotator through the exploration and the consolidation phase. By design, the platform can be used continuously, an annotator can go through exploration and consolidation in multiple iterations without losing progress.

For this study, we implement the modifications 1, 2, and 3, described in section 4.2.2, and we adapt the interface of the web platform. Together with additional explanations, the diagram from figure 4.3 is available to the annotator when using the web platform, serving as an instruction manual on how to perform the annotation correctly. Figures 5.1 and 5.2 on the next page show the interface of the web platform during exploration and consolidation.

The implementation of all pre-processing steps can be found in the repository [2] under *Value_Identification/pre-processing*. The adapted version of the Axies web platform can be found under *Value_Identification/axies*. The web platform is built with *Flask (version 2.1.3)*. Notably, it uses the Python framework *sentence-transformers (version 0.3.2)* to get the required sentence embedding models. All necessary dependencies are listed in the requirements file (*requirements.txt)*. The application can be deployed remotely or hosted locally.

[2] https://github.com/brigoraoul/VIVE



Welcome to exploration, [redacted]

The goal of the exploration phase is to derive a set of context-specific personal values, given a certain context. You will start with an empty list of values. You will be shown text messages and will be asked to annotate them as follows:

1. Add values: Can you identify an underlying personal value in the message? If a value comes to mind, can you complete the following sentence: "The author composed this message, because {value} is important to him/her.?" If yes, add the value to the list ("Add Value"). If the value is already in the list, select it ("Add value to message"). A value can be a single word or a term that contains multiple words.

2. Add keywords: When you add a new value to the list or select a value from the list, think whether there are any keywords for that value and add them ("Add keyword"). Keywords can be words that appear in the message or words that generally describe the value.

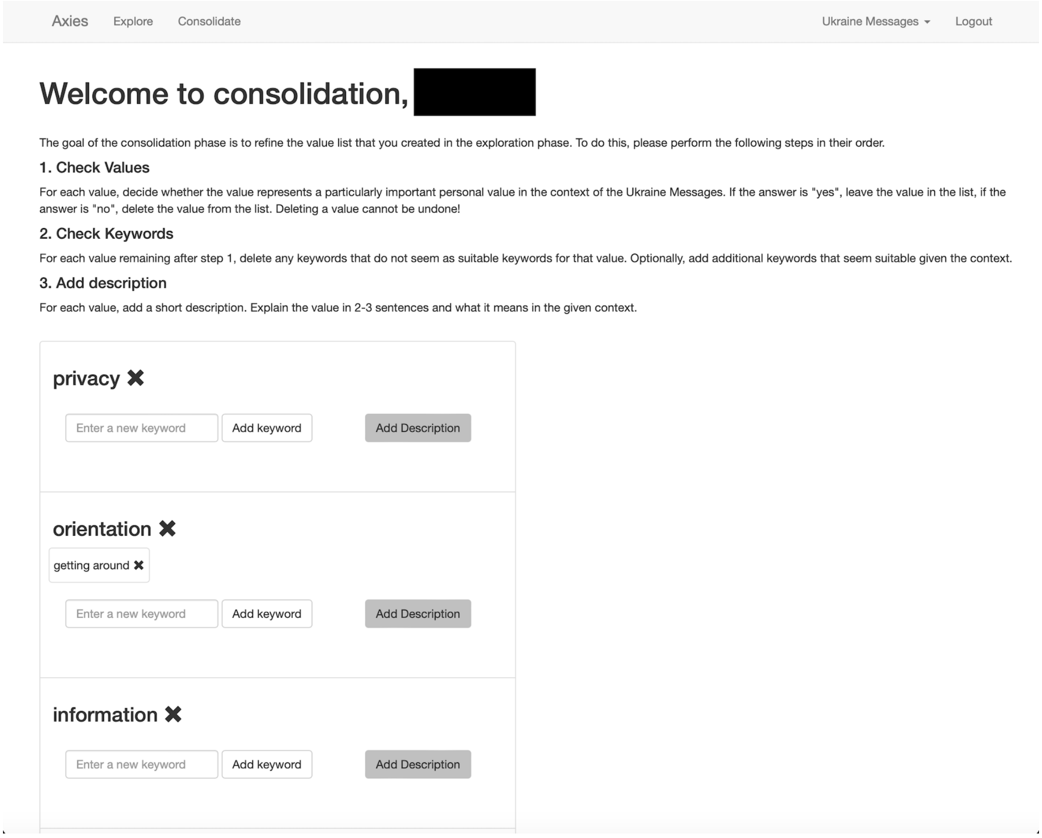
3. Related messages: When you add a new value, look at 2-3 potentially related messages ("Potential Related Message").

The exploration can be stopped when no new values are added after three or more consecutive new messages. However, at least 30 messages should be annotated with values. In case of doubt, please consult the exploration instructions.

Message:
Hello! Tell me, please, the address of the Red Cross and on what days does it work? Thank you very much.

What are the underlying personal values of this message?

FIGURE 5.1: Axies web platform exploration page



Welcome to consolidation, [redacted]

The goal of the consolidation phase is to refine the value list that you created in the exploration phase. To do this, please perform the following steps in their order.

1. Check Values

For each value, decide whether the value represents a particularly important personal value in the context of the Ukraine Messages. If the answer is "yes", leave the value in the list, if the answer is "no", delete the value from the list. Deleting a value cannot be undone!

2. Check Keywords

For each value remaining after step 1, delete any keywords that do not seem as suitable keywords for that value. Optionally, add additional keywords that seem suitable given the context.

3. Add description

For each value, add a short description. Explain the value in 2-3 sentences and what it means in the given context.

FIGURE 5.2: Axies web platform consolidation page

Value Representation

The implementation of the value representation module can be found in the repository under *Value_Representation*. It uses a custom Python class to store value objects (= value representations). In addition to the name, list of keywords, and description, value objects contain a unique identifier that makes it possible to store them in a dictionary. All information necessary for the creation of the value objects is queried from the database, using *SQLAlchemy (version 2.0.27)*.

Value Extraction

For this study, we developed a *Value Extraction Agent* (VEA) that performs value extraction as described in section 4.4. The implementation of the VEA can be found in the repository under *Value_Extraction/PersonalValueAgent*. Analogously to the value identification module, all necessary dependencies are listed in the requirements file (*requirements.txt*). In summary, the VEA is a standalone Python application that can be used for the task of value extraction.

Figure 5.3 shows a UML class diagram of the VEA. The agent is built following the standard principles of software design by Gamma et al. [23]. For the sake of clarity, the diagram in 5.3 only displays the classes and methods that implement essential parts of the VEA. The repository contains a more complete description of the implementation. As figure 5.3 shows, an instance of the personal value agent (or *ValueExtractionAgent*) comprises a list of value extraction sources, that are realizations of the interface *IValueExtractionSource*. For this study, we implement two types of value extraction sources - the concrete classes *Dictionary* and the *LLM*. However, the agent is designed to incorporate an arbitrary number of value extraction sources. For example, an additional machine learning classifier could be used for value extraction and could be added as another realization of *IValueExtractionSource*. The VEA allows the user to specify a set of value extraction sources. The specified value extraction sources are instantiated through pre-written modes that are defined in the class *EValueExtractionSource*. This architecture follows the strategy design pattern [23].

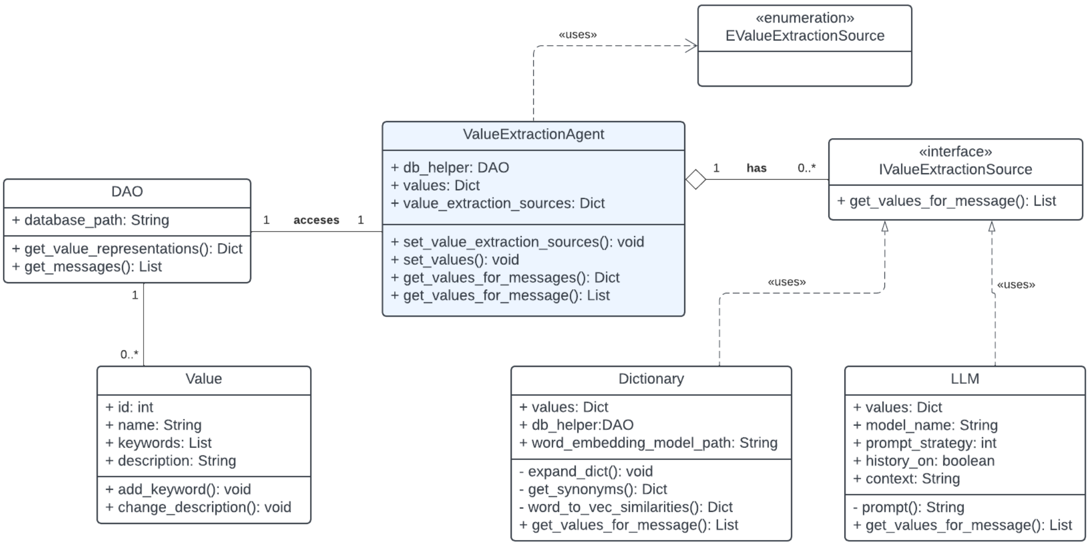


FIGURE 5.3: UML class diagram of the value extraction agent (VEA).

The VEA utilizes large language models via a local inference server from *Ollama (version 0.2.0)*. Ollama is an open-source project that provides an API to download and run quantized LLMs locally. We opted for quantized models from Ollama for the reasons explained in section 4.4.1. As mentioned in section 3.3.3, the dictionary is expanded with synonyms and word embedding similarities. Our implementation uses the *NLTK (version 3.8.1)* library [7] to obtain a list of synonyms for each of the initially present keywords. We use a Word2Vec embedding model from *Gensim (version 4.3.2)* [53], a Python library for topic modeling, to calculate the five most similar terms for each of the initially present keywords. The similarity between terms is defined as the distance in the semantic embedding space of the model.

5.3 Results

This section reports the results of the conducted experiments. It separates the results of the value identification and the results of the value extraction. This section does not contain the discussion of the results.

5.3.1 Value Identification

During the value identification via Axies for the Ukraine data set, a total of 40 messages were annotated with a total of 16 unique values. Out of the 40 messages, 18 were annotated as "Not Comprehensible", and were therefore not assigned to any values. Table 5.4 displays the names of all 16 identified, context-specific values and the message for which they were first annotated. Note that table 5.4 does not show the full value representations. A table containing the full value representation of all 16 identified personal values can be found in Appendix B.

TABLE 5.4: Context-specific personal values that were identified for the Ukraine data set.

Name	Message
shelter	Hello, is the program for free housing for Ukrainians still valid?
important parcel	Please help! Do you need an address to receive the package, who can accept the package? Please drop the postal address, please
mental health	Good afternoon. I am a practicing psychologist, family psychologist/psychotherapist. If you need individual or pair consultation, are confused in your life circumstances, I will be happy to help you find a way out together. Record in personal. May everything be fine with you
financial matters	Where is the money in the ATM of the private, the area of the covered - social city?
staying connected	"Heyou Tell me, please, where/to whom in Bratislava can the phone be taken for repair?"
staying warm in winter	Good evening .Please tell me the number of the master of gas columns
work	Good afternoon, who can help with the medical commission in Bucharest for seafarers?
getting around	Good evening) how to get from Bucharest to Chisinau? Can't find regular bus services
obeying traffic rules	Please tell me now you need a vinette for a car if you are in <PERSON>?
health	Please advise a Russian-speaking or Ukrainian speaking therapist for a teenager in Krakow
pet	Good <URL>ybe someone has guinea pigs?We would buy - you need a boy)).
everyday life	Hello! Tell me, please, is there a good master of household appliances? Preferably in the center. Thanks
translation services	Hello! Tell me, please, there is a bureau in Poland that translates a document immediately from Ukrainian into English, and not into Polish, and then into English?
important documents	Hello, friends. Who dealt with the registration of temporary guardianship for children? Where is this done and in what time frame?
disappointment in this city/country	I think we will go home and get help from the Red Cross there faster than in Warsaw.
help for refugees	Hello you can address the red cross and what help you can get there please tell me where you are in Bucharest organizations where they give help to children and mother

5.3.2 Value Extraction

This section reports the obtained results for the evaluation metrics from section 5.2.2 for all six classifiers from table 5.1. The subsequent sections contain discussions of these results with regard to the experimental research questions. As explained in section 5.2.2, the task of value extraction can be viewed as a multi-label or a single-label task. Accordingly, for each of the evaluated classifiers, we report separate results for the multi-label and the single-label task. When mentioning *accuracy*, we generally refer to the exact match ratio (MR).

Table 5.6 displays the accuracy, partial correctness (PC), precision, recall, and F1-score over all labels for the multi-label task. Analogously to table 5.6, except for the partial correctness, table 5.7 reports the metrics for the single-label task. All reported values represent the calculated mean over five independent trials. This means the experiment was conducted five times and the reported values are the average of the values obtained for each of the times. We report the average over five trials to reduce the impact of anomalies and randomness and provide more reliable results. Thus, the tables 5.6 and 5.7 also include the corresponding standard deviation (SD) for all reported values. Note that except for the rare case (4.118 percent of all cases in this experimental evaluation) that the dictionary extracts more than one value and picks a random value from the set of extracted values (see algorithm 5) the classifier *Dict* is deterministic. Hence, the standard deviation of the classifier *Dict* is zero for all metrics for the multi-label task, and for the single-label task, the standard deviations are much smaller than for the other classifiers. The tables illustrate the comparison of all six evaluated classifiers. The classifiers are listed horizontally, therefore the highest value of each row refers to the best-performing classifier according to the metric of that row. For each metric, the respective best-performing classifier is indicated with a blue coloring and the worst-performing classifier with a red coloring. For example, the highest value of the first row in table 5.6 is *0.527*. The cell that contains the value *0.527* is located in the column *llama3* and in the row *Accuracy*. Hence, the classifier *llama3* is the best-performing classifier in terms of accuracy (exact match ratio) for the multi-label task.

Table 5.8 reports the precision, recall, and F1-score for each of the five examined personal values (see table 5.3) individually for the multi-label task. Analogously, table 5.9 reports the metrics for the single-label task. Both tables report values for all six evaluated classifiers. In addition to the coloring scheme from the tables 5.6 and 5.7, a blue dot ● indicates the personal value for which a classifier achieved the best result for a certain metric. The tables in this section that report evaluation metrics for each of the five personal values from table 5.3 individually, namely 5.8 and 5.9, do not report the accuracy per value, for the reason explained in 5.2.2.

Figures 5.4 to 5.9 show the confusion matrices for the single-label task for all six evaluated classifiers. They display six classes because in addition to the five personal values from table 5.5 the class "no label" is included, representing messages that do not reference any one of the five values. The numbers on the axis refer to the value IDs from table 5.5. The class "no label" is assigned number 5. All confusion matrices display the sum of five confusion matrices that were obtained over five independent trials.

TABLE 5.5: Numbering of personal values for evaluation

Value ID	Value Name
0	shelter
1	mental health
2	staying connected
3	disappointed in this city/country
4	help for refugees
5	no label

TABLE 5.6: Multi-label message classification results

Metric	Dict		llama3		Dict+llama3		mistral		gemma		Simple repr.	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Accuracy	0.3824	0.0	0.527	0.014	0.4576	0.016	0.5188	0.016	0.5094	0.014	0.4569	0.009
PC	0.8814	0.0	0.8989	0.013	0.8756	0.017	0.9072	0.017	0.9137	0.019	0.8974	0.013
Precision	0.5821	0.0	0.6228	0.006	0.5411	0.011	0.6186	0.007	0.6081	0.006	0.6148	0.011
Recall	0.2955	0.0	0.7227	0.017	0.7864	0.028	0.7424	0.024	0.7424	0.016	0.6112	0.021
F1	0.392	0.0	0.6689	0.007	0.641	0.017	0.6748	0.014	0.6611	0.009	0.613	0.008

TABLE 5.7: Single-label message classification results

Metric	Dict		llama3		Dict+llama3		mistral		gemma		Simple repr.	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Accuracy	0.3964	0.0001	0.7259	0.014	0.6058	0.013	0.7294	0.001	0.74	0.013	0.6435	0.012
Precision	0.3988	0.0	0.7937	0.013	0.6449	0.007	0.7894	0.009	0.8018	0.007	0.7818	0.009
Recall	0.2548	0.0001	0.8238	0.011	0.8305	0.007	0.827	0.009	0.8285	0.019	0.6587	0.014
F1	0.3109	0.0001	0.8085	0.011	0.726	0.007	0.8078	0.008	0.8149	0.011	0.715	0.013

TABLE 5.8: Multi-label message classification results per value. A blue dot indicates the value for which a classifier achieved the best result per metric.

Value	Metric	Dict		llama3		Dict+ llama3		mistral		gemma		Simple repr.	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
shelter	Precision	0.4211	0.0	0.7272	0.024	0.4821	0.020	0.7105	0.012	0.7143	0.014	0.875	0.016
	Recall	0.5517	0.0	0.8276 ●	0.019	0.931 ●	0.022	0.931 ●	0.013	0.8621 ●	0.018	0.7241	0.017
	F1	0.4776	0.0	0.7742	0.015	0.6353	0.013	0.806	0.010	0.7813	0.022	0.7925	0.016
mental health	Precision	1.0 ●	0.0	0.88	0.012	0.92	0.018	0.9615	0.011	1.0 ●	0.014	1.0 ●	0.023
	Recall	0.0357	0.0	0.7857	0.016	0.8214	0.022	0.8928	0.010	0.8214	0.018	0.6786	0.014
	F1	0.069	0.0	0.8302	0.017	0.868 ●	0.022	0.9259 ●	0.009	0.902 ●	0.012	0.8085 ●	0.014
staying connected	Precision	0.7826	0.0	0.92	0.012	0.8	0.019	0.8846	0.011	0.8846	0.014	0.9375	0.024
	Recall	0.6207 ●	0.0	0.7931	0.021	0.8275	0.013	0.7931	0.011	0.7931	0.015	0.5172	0.012
	F1	0.6923 ●	0.0	0.8519 ●	0.015	0.8136	0.022	0.8364	0.013	0.8364	0.011	0.6667	0.014
disappointment in this city/country	Precision	0.0	0.0	1.0 ●	0.017	1.0 ●	0.013	1.0 ●	0.012	0.75	0.023	1.0 ●	0.019
	Recall	0.0	0.0	0.25	0.011	0.3333	0.014	0.3333	0.023	0.25	0.015	0.1666	0.008
	F1	Inf	0.0	0.4	0.019	0.5	0.015	0.5	0.020	0.375	0.022	0.2857	0.012
help for refugees	Precision	0.8	0.0	0.3235	0.022	0.2877	0.010	0.3088	0.018	0.3143	0.012	0.3472	0.016
	Recall	0.1176	0.0	0.6176	0.020	0.6471	0.016	0.6176	0.012	0.6471	0.011	0.7353 ●	0.019
	F1	0.2051	0.0	0.4246	0.019	0.3983	0.022	0.4118	0.015	0.423	0.017	0.4717	0.012

TABLE 5.9: Single-label message classification results per value. A blue dot indicates the value for which a classifier achieved the best result per metric.

Value	Metric	Dict		llama3		Dict+ llama3		mistral		gemma		Simple repr.	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
shelter	Precision	0.4211	0.0	0.8889	0.018	0.5094	0.009	0.8065	0.024	0.8929	0.017	0.7586	0.020
	Recall	0.5714	0.0	0.8571	0.014	0.9259	0.019	0.8928	0.011	0.8928	0.021	0.7857	0.023
	F1	0.4848	0.0	0.8727	0.015	0.6572	0.010	0.8474	0.008	0.8928	0.019	0.7719	0.012
mental health	Precision	1.0 ●	0.0	0.931	0.020	0.9	0.018	0.9259	0.019	0.931	0.013	1.0 ●	0.017
	Recall	0.037	0.0	1.0 ●	0.011	1.0 ●	0.020	0.9259	0.018	1.0 ●	0.014	0.8148 ●	0.019
	F1	0.0714	0.0	0.9643 ●	0.018	0.9474 ●	0.013	0.9259	0.011	0.9643 ●	0.016	0.898 ●	0.022
staying connected	Precision	0.7906	0.0001	0.9259	0.011	0.792	0.023	0.9643	0.017	0.9643	0.019	0.909	0.012
	Recall	0.6107 ●	0.0001	0.862	0.008	0.8655	0.014	0.931 ●	0.015	0.931	0.010	0.6897	0.021
	F1	0.6891 ●	0.0001	0.8928	0.012	0.8271	0.011	0.9474 ●	0.014	0.9474	0.018	0.7843	0.017
disappointment in this city/country	Precision	0.0	0.0	1.0 ●	0.015	1.0 ●	0.022	1.0 ●	0.017	1.0 ●	0.009	1.0 ●	0.013
	Recall	0.0	0.0	0.3182	0.010	0.3636	0.012	0.3636	0.017	0.2727	0.011	0.1818	0.009
	F1	Inf	0.0	0.4828	0.012	0.5333	0.014	0.5333	0.018	0.4286	0.016	0.3077	0.012
help for refugees	Precision	0.8	0.0	0.5789	0.013	0.5714	0.017	0.5238	0.019	0.5435	0.016	0.52	0.011
	Recall	0.129	0.0	0.7096	0.014	0.7742	0.011	0.7097	0.021	0.8065	0.013	0.4194	0.020
	F1	0.2286	0.0	0.6376	0.010	0.6575	0.012	0.6027	0.014	0.6494	0.016	0.4643	0.011

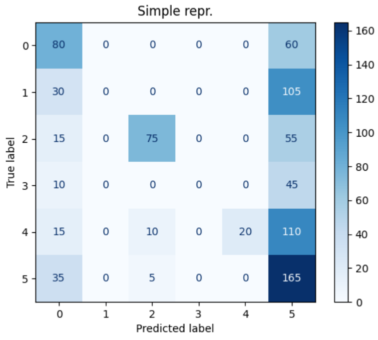


FIGURE 5.4: Confusion matrix for the single-label task, using the dictionary.

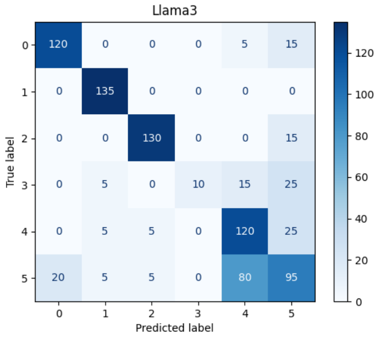


FIGURE 5.5: Confusion matrix for the single-label task, using Llama3.

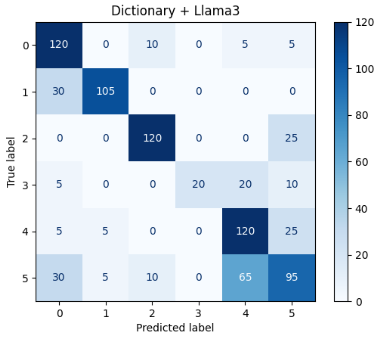


FIGURE 5.6: Confusion matrix for the single-label task, using the dictionary and Llama3.

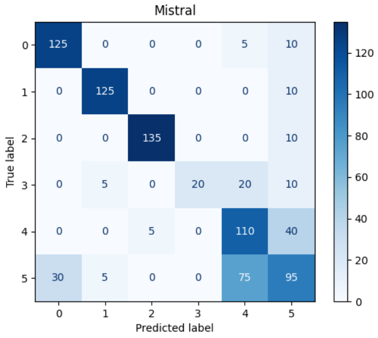


FIGURE 5.7: Confusion matrix for the single-label task, using the Mistral model.

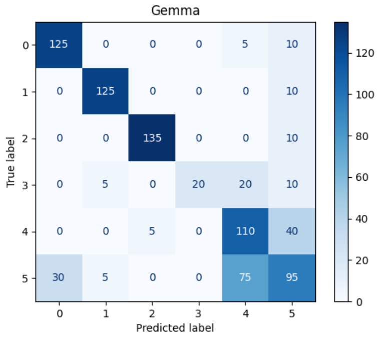


FIGURE 5.8: Confusion matrix for the single-label task, using the Gemma model.

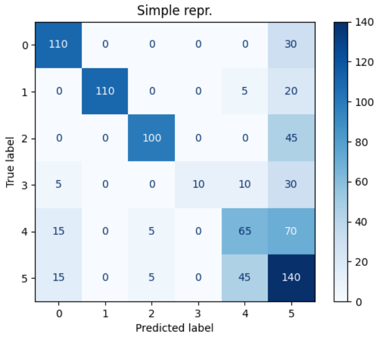


FIGURE 5.9: Confusion matrix for the single-label task, using a simple value representation.

5.4 Combining LLMs and a Dictionary

The purpose of experimental research question 1 of this study is to test the hypothesis that a combination of LLMs and a dictionary for value extraction leads to significantly higher accuracy compared to using either LLMs or a dictionary alone. To test this hypothesis we compare the three classifiers *Dict*, *llama3*, and *Dict+llama3*.

Discussion of Evaluation Metrics From tables 5.6 and 5.7, we can see that the *Dict* classifier has the worst performance according to almost all evaluation metrics. The only exceptions are the PC and the precision for the multi-label task, for which *Dict+llama3* performs slightly worse than *Dict*. *llama3* performs better than the *Dict* for all metrics. For example, the single-label accuracy of *llama3* is 0.7259 (±0.014), compared to 0.3941 (±0.0) for *Dict*. *llama3* also performs better than *Dict+llama3* for all metrics with the exception of recall. *Dict+llama3* yields a higher recall compared to *llama3* both, for the multi-label task (+0.0637) and the single-label task (+0.019).

These results indicate that using only LLMs leads to better performance for value extraction than using only a dictionary or a combination of a dictionary and LLMs. The higher recall of *Dict+llama3* compared to *llama3* was expected, given the way our value extraction agent (see section 5.2.3) combines the individual outputs of all value extraction sources. As explained in section 3.3.4, the agent optimizes for recall by taking the union of the values extracted by the dictionary and the values extracted by LLMs. Therefore, and because the calculated metrics for the three classifiers are obtained over the same trials, the recall of *Dict+llama3* must by design be higher than or at least equal to the recall of both *Dict* and *llama3*. This also applies to the recall per value that is shown in table 5.8.

Figure 5.4 shows the confusion matrix for the *Dict* classifier. It shows that *Dict* yields a high number of false negatives for all values. Meaning, that regardless of the personal value that is annotated for a message, there is a high chance that *Dict* extracts no value. For example, from the second top row, which refers to the value *mental health* we can see that 105 out of 135 instances are classified as "no label". Because the confusion matrices display the sum over five trials, this translates to an average of 21 out of 27 messages referencing the value *mental health* per trial. These findings indicate that the dictionary is not powerful enough. Generally, the more values a dictionary extracts for a given natural language data set, the more powerful it is. Section 3.3.3 elaborates on the meaning of the term *powerful*. We identify the "powerfulness" of the dictionary as a limitation of the dictionary approach for value extraction. In simple terms, we conclude from the results that the dictionary extracts too few values because it contains too few keywords. As a consequence, it negatively influences the value extraction of *Dict+llama3*.

Statistical Analysis We conduct a statistical comparison of the *Dict*, *llama3*, and *Dict+llama3*, following the method explained in section 5.1. Because we cannot assume normal distribution and variance of the data we use the non-parametric Friedman test as an omnibus test to check for differences in the multi-label accuracy (Multi-Acc.), partial correctness (PC), and single-label accuracy (Single-Acc.). For all three compared metrics we obtain the same result. The null hypothesis of the Friedman test that there is no difference between the distributions is rejected. Therefore, we further explore the present difference with the post-hoc Nemenyi test as proposed by Demšar et al. [16]. The Nemenyi test compares all three classifiers pairwise based on the absolute differences in their average rankings. It determines a critical difference (CD) and has the null hypothesis that all classifiers have equal performance. If the average ranking difference is greater than the critical difference, the null hypothesis is rejected. The below figures show these pairwise differences for the comparison of the Multi-Acc. (figure 5.10), the PC (figure 5.11) and the Single-Acc. (figure 5.12). Two classifiers that are connected by a horizontal bar, are considered not significantly different in their performance [30]. For example, from figure 5.10 we see that the classifiers *Dict* and *Dict+llama3* as well as *Dict+llama3* and *llama3* are not significantly different. In contrast, *Dict* and *llama3* appear to be significantly different. The Nemenyi null hypothesis is rejected for the Multi-Acc at a p-value of 0.0067, for the PC at a p-value of 0.0907, and for the Single-Acc likewise at a p-value of 0.0067. This is based on a calculated *CD* of 1.4823 for a confidence value $\alpha = 0.05$. Table 5.10 summarizes the statistical comparison by displaying the mean, standard deviation (SD), range between the lowest and highest observed value (d), and effect size (Magnitude). The effect size is calculated using Cohen's d [58] and is a measure of the size and practical importance of the effect.

Metric	Classifier	Mean	SD	CI	d	Magnitude	
Single-Acc.	Dict	3.000	0.396	0.000	[0.388, 0.417]	-20.571	large
	Dict+llama3	2.000	0.606	0.015	[0.598, 0.629]	-24.002	large
	llama3	1.000	0.726	0.015	[0.699, 0.753]	-31.672	large
Multi-Acc.	Dict	3.000	0.382	0.000	[0.382, 0.382]	-	large
	Dict+llama3	2.000	0.458	0.018	[0.425, 0.490]	-5.812	large
	llama3	1.000	0.527	0.015	[0.501, 0.553]	-13.861	large
PC	Dict	2.400	0.881	0.000	[0.881, 0.881]		
	Dict+llama3	2.400	0.876	0.019	[0.841, 0.910]		
	llama3	1.200	0.899	0.015	[0.873, 0.925]		

TABLE 5.10: Summary of the statistical comparison of *Dict*, *llama3*, and *Dict+llama3*

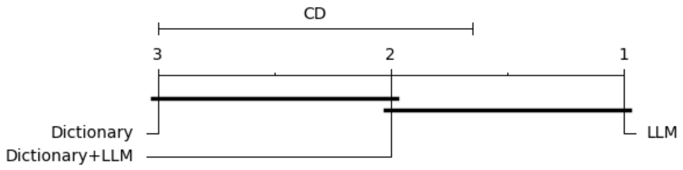


FIGURE 5.10: Pairwise accuracy comparison for the multi-label task.

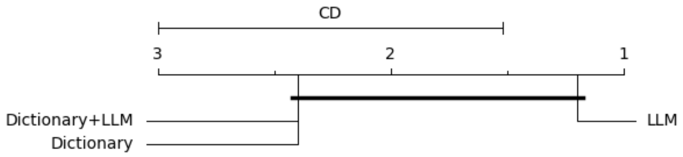


FIGURE 5.11: Pairwise partial accuracy comparison for the multi-label task.

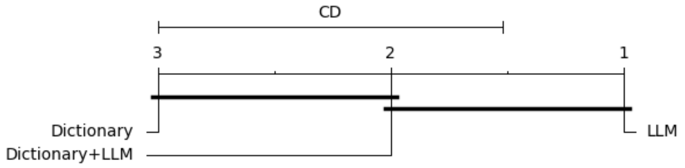


FIGURE 5.12: Pairwise accuracy comparison for the single-label task.

Conclusion Based on the above-presented analysis we reject hypothesis 1 for experimental research question 1. We conclude that the combination of Llama3, representative of LLMs, and a dictionary does not improve the accuracy of the value extraction, given the way the dictionary is expanded (see section 3.3.3), the way the LLM is prompted (see section 4.4.1), and the way the value extraction module of VIVE is implemented for this study (see section 5.2.3). The main reason for this conclusion is that there is no significant improvement in accuracy when combining Llama3 and a dictionary. In fact, the accuracy of *llama3* is significantly higher than the accuracy of both *Dict* and *Dict+llama3*. Furthermore, *llama3* outperforms *Dict* and *Dict+llama3* in terms of precision, recall, and F1, both on average and per value for almost all values.

5.5 Using different large language models

Experimental research question 2 essentially asks whether the choice of LLM has a significant impact on the performance of value extraction. To test our hypothesis that the use of different LLMs does not lead to significantly different accuracies, we compare the three classifiers *llama3*, *mistral*, and *gemma*. Based on the result from experiment 1 (see section 5.4) that *llama3* outperforms *Dict+llama3*, we do not include the dictionary in this experiment. For readability purposes, table 5.11 combines the overall results of the classifiers *llama3*, *mistral*, and *gemma* for both the multi-label and the single-label value extraction task. These results are also displayed in tables 5.6 and 5.7.

TABLE 5.11: Comparison of large language models

Task	Metric	llama3		mistral		gemma	
		Mean	SD	Mean	SD	Mean	SD
Multi-label	Accuracy	0.527	0.014	0.5188	0.016	0.5094	0.014
	PC	0.8989	0.013	0.9072	0.017	0.9137	0.019
	Precision	0.6228	0.006	0.6186	0.007	0.6081	0.006
	Recall	0.7227	0.017	0.7424	0.024	0.7424	0.016
	F1	0.6689	0.007	0.6748	0.014	0.6611	0.009
Single-label	Accuracy	0.7259	0.014	0.7294	0.001	0.74	0.013
	Precision	0.7937	0.013	0.7894	0.009	0.8018	0.007
	Recall	0.8238	0.011	0.827	0.009	0.8285	0.019
	F1	0.8085	0.011	0.8078	0.008	0.8149	0.011

Discussion of Evaluation Metrics Table 5.11 shows that for the multi-label classification task *llama3* achieves the highest accuracy out of all evaluated classifiers, with

$MR = 0.527$, which is slightly higher than *mistral* (+0.0082) and *gemma* (+0.0176). *llama3* also achieves the highest precision, with $P = 0.6228$. However, when predicting individual values (PC), we observe the best performance for *gemma*, with $PC = 0.9137$. We observe the highest F1 score for *mistral*, with $F1 = 0.6748$. These results indicate that the three classifiers *llama3*, *mistral*, and *gemma* all have slightly different strengths. However, it has to be noted that the differences for all metrics are marginal. A quantification of the differences is provided by the statistical analysis below. Table 5.11 shows that in the single-label classification task *gemma* outperforms *llama3* and *mistral* in all evaluation metrics. In terms of accuracy, precision, and the F1 score, *gemma* outperforms all evaluated classifiers. Only in terms of recall, *gemma* performs worse (-0.0143) than *Dict+llama3*.

Statistical Analysis We conduct a statistical comparison of *llama3*, *mistral*, and *gemma*, analogously to the statistical comparison described in section 5.4, using the Friedman test as an omnibus test. For the multi-label accuracy ($p = 0.152$), partial correctness ($p = 0.2105$), and single-label accuracy ($p = 0.4025$), the null hypothesis is not rejected. Therefore, we conclude that there is no significant difference in performance between *llama3*, *mistral*, and *gemma* and we do not perform a post-hoc test. Figures 5.13, 5.14, and 5.15 show the confidence intervals per classifier for a confidence value $\alpha = 0.05$. For all three metrics, the confidence ranges of all three classifiers overlap, illustrating that there is no significant difference.

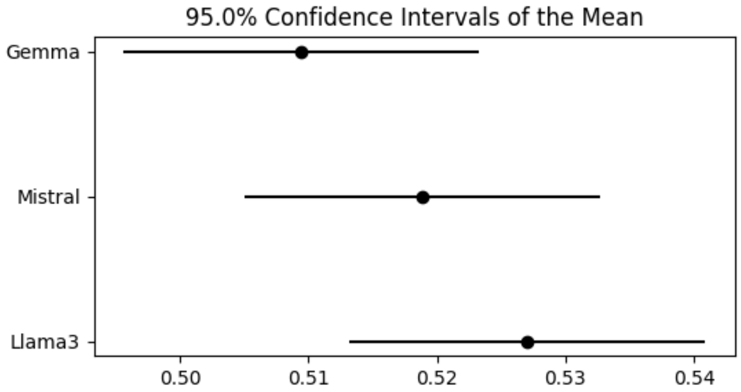


FIGURE 5.13: Pairwise accuracy comparison for the multi-label task.

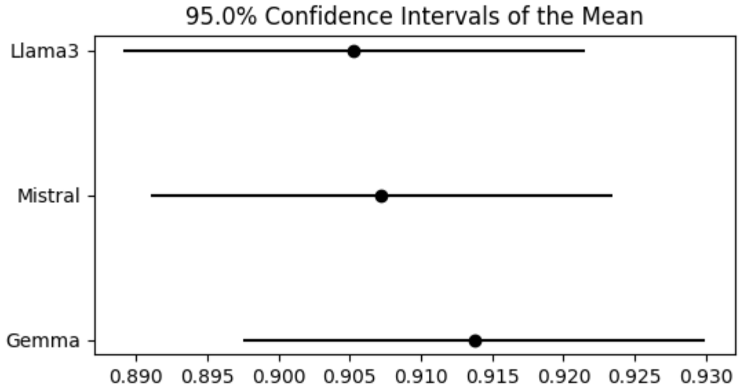


FIGURE 5.14: Pairwise partial accuracy comparison for the multi-label task.

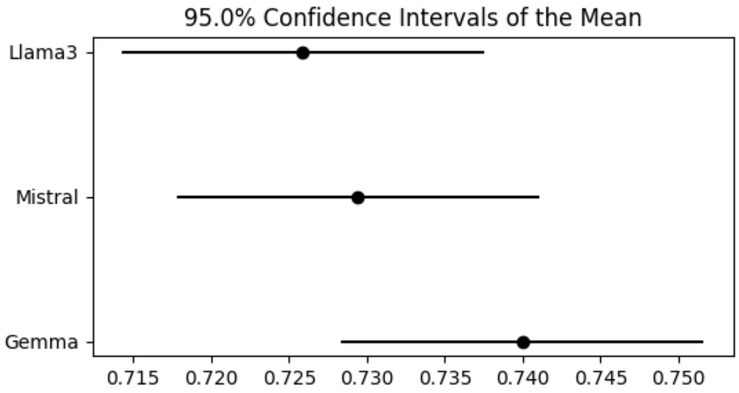


FIGURE 5.15: Pairwise accuracy comparison for the single-label task.

Conclusion Based on the above-presented analysis we do not reject hypothesis 2 for experimental research question 2. We conclude that using any of the three utilized LLMs — Llama3, Mistral, or Gemma — does not significantly impact the accuracy of value extraction. Further, we conclude that all three utilized LLMs can effectively extract personal values from natural language text because they achieve accuracies that are significantly higher than random chance.

5.6 Using different value representations

To answer experimental research question 3, we compare two versions of the classifier *llama3*. *llama3* is the best-found classifier from experiment 1 (see section 5.4). In the preceding experiment 2 (see section 5.5), no other classifier is shown to significantly outperform *llama3*, which is why we keep it as (one of) the best-found classifiers for experiment 3. In this section, we refer to version 1 of *llama3*, as *Full repr.*, because it uses full value representations. We refer to version 2 of *llama3* as *Simple repr.*, because it uses simplified value representations. For readability purposes, table 5.12 combines the overall results of the classifiers *Full repr.* and *Simple repr.* for both the multi-label and the single-label value extraction task.

TABLE 5.12: Comparison of value representations

Task	Metric	Full repr.		Simple repr.	
		Mean	SD	Mean	SD
Multi-label	Accuracy	0.527	0.014	0.4569	0.009
	PC	0.8989	0.013	0.8974	0.013
	Precision	0.6228	0.006	0.6148	0.011
	Recall	0.7227	0.017	0.6112	0.021
	F1	0.6689	0.007	0.613	0.008
Single-label	Accuracy	0.7259	0.014	0.6435	0.012
	Precision	0.7937	0.013	0.7818	0.009
	Recall	0.8238	0.011	0.6587	0.014
	F1	0.8085	0.011	0.715	0.013

Discussion of Evaluation Metrics As table 5.12 shows, *Full repr.* outperforms *Simple repr.* in all metrics. This suggests that *Full repr.* is in every way the better classifier. However, tables 5.8 and 5.9 allow a more in-depth analysis. For the single-label task (5.9), *Full repr.* outperforms *Simple repr.* in terms of precision, for all values except *mental health*. In contrast, for the multi-label task (5.8), *Simple repr.* outperforms *Full repr.* in terms of precision per value, for all values. This represents an instance of the Simpson's paradox. Simpson's paradox occurs when the aggregate-level observation indicates that classifier A performs better, but when examining individual categories, classifier B performs better. A frequent cause of Simpson's paradox is a "hidden" variable that influences the relationship between the measured metrics. A possible explanation is the value specificity. *Simple repr.* might be better at extracting more concrete and less specific values, like *shelter* or *mental health*, whereas *Full repr.* might be better suited for more abstract and more specific values, like *disappointment in this city/country* or *help for refugees*. However, developing an exact measure for value specificity exceeds the scope of this study.

Statistical Analysis Because we compare only two classifiers in this experiment, an omnibus test is not required. Instead, we compare *Full repr.* and *Simple repr.* through paired t-tests. The null hypothesis that there is no difference between the distributions is rejected for the multi-label accuracy (Multi-Acc.) at a p-value of 0.0005 and the single-label accuracy (Single-Acc.) at a p-value of 0.0004. The null hypothesis is not rejected for the partial correctness (PC) at a p-value of 0.3965. We conclude that *Full repr.* performs significantly better than *Simple repr.* in terms of exact match accuracies (Multi-Acc. and Single-Acc.). We further conclude that there is no significant difference in terms of predicting individual values (PC). The latter implies that given a message d and a value v, the sole decision of whether d references v is not significantly impacted by the value representation. However, to extract the correct subset $V_d \subseteq V_{c,A}$, a more comprehensive value representation seems to achieve a better result. Table 5.13 gives an overview of the statistical comparison between *Full repr.* and *Simple repr.* by displaying the mean, standard deviation (SD), range between the lowest and highest observed value (d), and effect size (Magnitude). Figures 5.16, 5.17, and 5.18 illustrate the result for confidence value $\alpha = 0.05$. In figure 5.16 and 5.18, the two black, horizontal lines do not overlap, indicating a significant difference.

Metric	Classifier	Mean	SD	d	Magnitude
Multi-Acc	Simple Repr	0.455	0.013	[0.435, 0.475]	negligible
	Full Repr.	0.527	0.015	[0.504, 0.550]	large
PC	Simple Repr	0.891	0.014	[0.869, 0.912]	
	Full Repr.	0.899	0.015	[0.876, 0.922]	
Single-Acc	Simple Repr	0.644	0.014	[0.622, 0.665]	negligible
	Full Repr.	0.726	0.015	[0.702, 0.750]	large

TABLE 5.13: Single-label accuracy for different representation

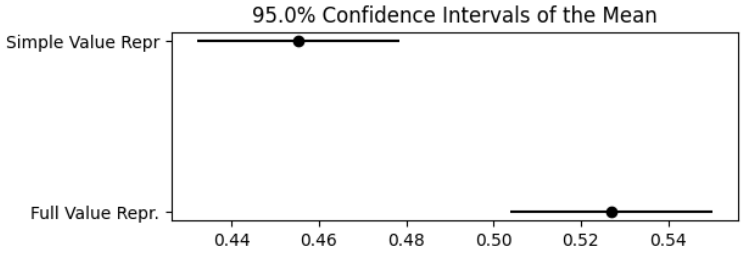


FIGURE 5.16: Pairwise accuracy comparison for the multi-label task.

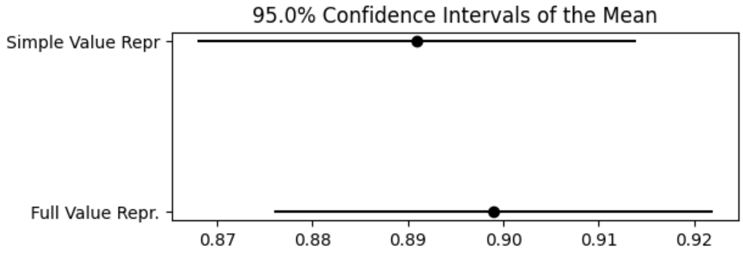


FIGURE 5.17: Pairwise partial accuracy comparison for the multi-label task.

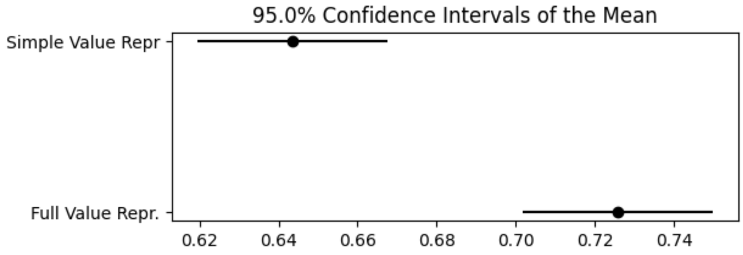


FIGURE 5.18: Pairwise accuracy comparison for the single-label task.

Conclusion Based on the above-presented analysis we do not reject hypothesis 3 for experimental research question 3. We conclude that the utilized value representation significantly impacts the accuracy of the value extraction. Considering the two value representations that we use in this experiment, the results validate our assumption that a more extensive value representation is beneficial for value extraction via LLMs.

5.7 Usefulness Evaluation

To further address sub-research question 4, *"What is the accuracy, precision, recall, and usefulness of the proposed method for value extraction?"* (see section 1.5.2), we conduct a user study with three analysts from the Netherlands Red Cross. The user study aims to evaluate the usefulness of VIVE in the context of analyzing feedback data from humanitarian programs, such as the Ukraine data set. As explained in section 1.5.2, we regard usefulness as a measure of how practical and valuable the output of VIVE is to the user.

For this user study, we extract values for all 4522 messages from the Ukraine data set, using the best-found classifier *llama3*. Figures 5.19 and 5.20 show the frequency of the 16 identified context-specific personal values (see table 5.4) in the Ukraine data set. The charts from the figures 5.19 and 5.20 are included in an exemplary output document of VIVE, that we designed for this user study. Together with a brief explanation of the purpose of VIVE, the exemplary VIVE output document is presented to the participants of the user study. The participants are then presented with six statements about the usefulness of VIVE and are asked to indicate on a Likert Scale [34], how strongly they agree or disagree with the statements. The user study questionnaire and the exemplary VIVE output document are available as a PDF and can be found in this project's GitHub repository[3], under *Netherlands Red Cross Case Study/User Study*.

Table 5.14 shows the five options that participants can select from to indicate their agreement with a statement. Table 5.14 also shows a corresponding agreement score for each option. The agreement scores represent a numerical indication of how much a participant agrees with a statement, with a higher score indicating a higher agreement. Besides choosing one of the options from table 5.14, participants have the option to write a comment for each of the statements.

TABLE 5.14: User study selection options

Option	Score
Strongly disagree	1
Somewhat disagree	2
Neither agree nor disagree	3
Somewhat agree	4
Strongly agree	5

Table 5.15 presents the results of our user study. For each statement, it shows the indicated agreement per participant and the average agreement. The numbers in table 5.15 refer to the agreement scores. As table 5.15 shows, the average agreement is at least 4 for all statements, with the highest average agreement being 5 for statement 5: *The information contained in the report generated by VIVE is clear.* Statements 1 and 2 pertain to the requirements 1 and 2 from our problem investigation. The high agreement with statements 1 and 2 indicates that VIVE indeed addresses the requirements of the Netherlands Red Cross for the automated processing of feedback data. However, to further examine how well VIVE addresses the requirements, a treatment implementation and corresponding evaluation, as described by the engineering cycle from Wieringa et al.[75], is required. This exceeds the scope of this study. One participant commented in response to statement 3: *...the only missing*

[3] `https://github.com/brigoraoul/VIVE`

piece is some interactivity to enable the user to "drill down" specific issues. This would require more elaborate reporting of the results of VIVE than we performed for this user study. Section 6.2.1 describes a dedicated value reporting module as a possible extension of VIVE to address requirements such as interactivity of the VIVE output report.

TABLE 5.15: User study agreement scores

ID	**Statement**	**Participant 1**	**Participant 2**	**Participant 3**	**Average**
1	Having a tool like VIVE for Value Identification and Value Extraction is useful to automatically extract information from feedback data (e.g. the Ukraine Telegram messages) that is relevant to humanitarian programs.	4	5	4	4.33
2	Having a tool like VIVE for Value Identification and Value Extraction is useful to gain insights into individuals' needs and experiences, enabling the design of humanitarian programs with personalized support and improved communication.	4	5	5	4.67
3	The report generated by VIVE for the Ukraine Telegram messages is useful to make decisions about the design of humanitarian programs.	4	4	4	4
4	The information contained in the report generated by VIVE messages is comprehensive.	4	4	5	4.33
5	The information contained in the report generated by VIVE is clear.	5	5	5	5
6	The personal values that VIVE extracts for the example messages included in the report are accurate. The extracted personal values are actually underlying values of the message.	4	4	5	4.67

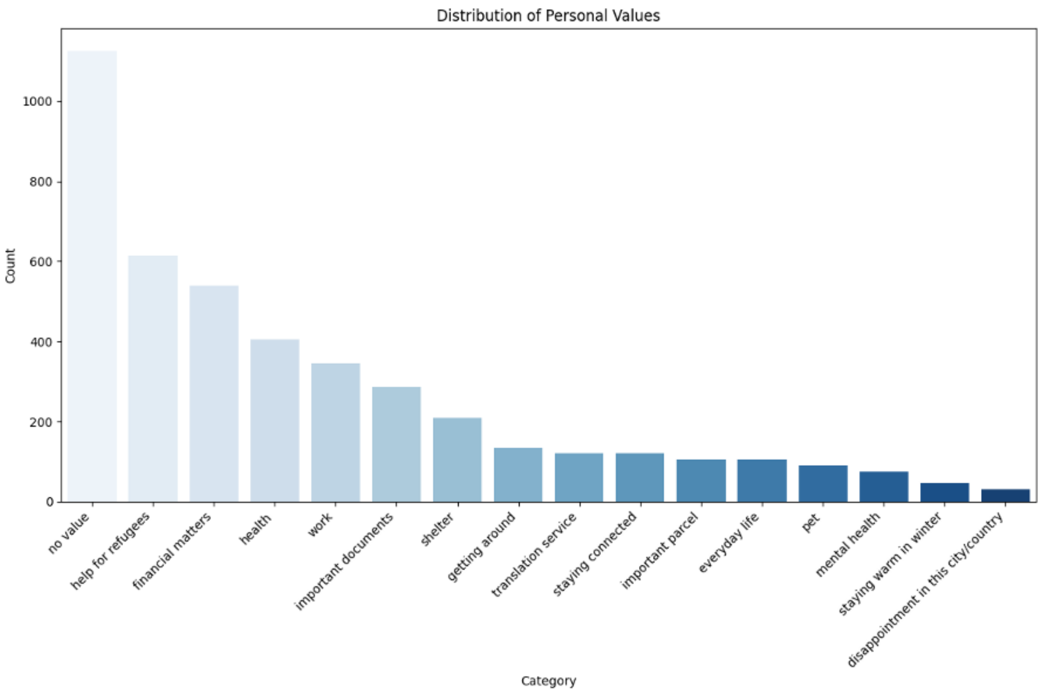


FIGURE 5.19: Distribution of the identified context-specific personal values over the Ukraine data set.

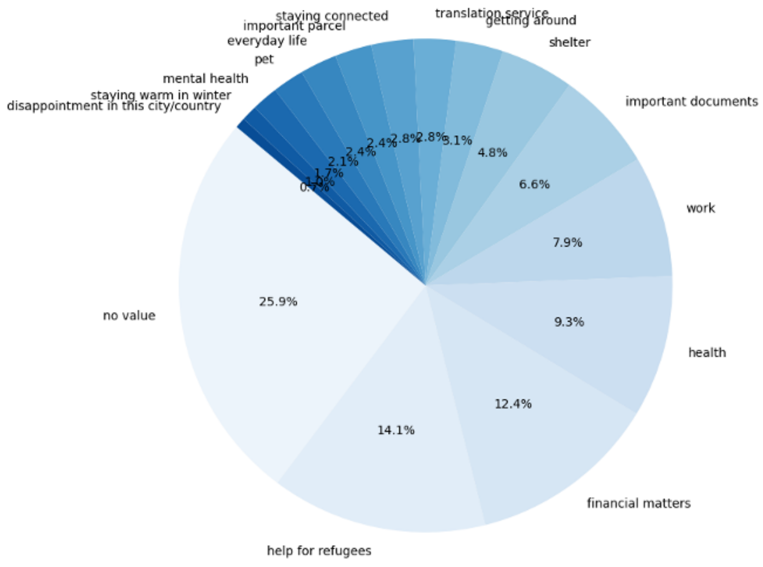


FIGURE 5.20: Distribution of the identified context-specific personal values over the Ukraine data set.

Chapter 6

Discussion

In this study, we propose VIVE - an end-to-end pipeline for the identification and extraction of context-specific values. The goal of this study is to design a tool using artificial intelligence to analyze feedback data from humanitarian projects in a way that supports the decision-making of humanitarian organizations, like the Netherlands Red Cross. Following the Design Science Methodology by Wieringa et al. [75] (see section 1.3), VIVE constitutes the artifact that addresses the requirements and limitations identified in our problem investigation. Namely, it addresses the requirements of the Netherlands Red Cross to automate the processing of humanitarian feedback data (see requirement 1) and to understand the experience of people in humanitarian crisis situations (see requirement 2). VIVE also addresses: 1) The lack of an LLM-based method for value extraction (see limitation 1), that exists in the literature despite the superior natural language understanding of LLMs compared to other natural language processing methods. 2) The lack of an end-to-end method for the identification and extraction of personal values (see limitation 2). Given a natural language data set that pertains to a certain context, VIVE can identify a set of context-specific personal values, represent these values computationally, and extract them from the individual texts in the data set. To demonstrate how to apply VIVE, we instantiate it for the context of the Ukraine data set (see section 4).

In this section, we further discuss the results of our evaluation of VIVE from section 5, highlighting the implications and limitations of this study. Building on the results, section 6.1 reflects on the research questions of this study. Section 6.2 describes possible extensions and future directions of research.

6.1 Reflection on Research Questions

Main research question *How to automatically extract context-specific personal values from natural language text?* We address the main research question by proposing VIVE. VIVE performs value extraction through a combination of a dictionary and large language models (LLMs). Following our design of the VIVE value extraction module, as described in section 3.3, values can be extracted via LLM inference-based algorithms (see algorithms 2 and 3) and via dictionary-based algorithms (see algorithms 4 and 5). We consider both, the LLM inference-based and the dictionary-based algorithms, as separate value extraction sources that provide an answer to sub-research question 3: *How to determine for a piece of text $d \in D_c$ and a context-specific value $v \in V_c$, whether v is referenced in d?* In principle, the outputs of different value extraction sources can be combined. This requires an algorithm that, for a given text d, combines the individual outputs of $\{V_{d,S_1}, V_{d,S_2}, \ldots, V_{d,S_n}\}$ from value extraction sources $\{S_1, S_2, \ldots, S_n\}$ to obtain the combined set of referenced values V_d. The algorithms 6 and 7 are examples of such an algorithm for the combination of different value extraction sources.

When aiming to increase the accuracy, precision, or recall of the value extraction it only makes sense to combine different value extraction sources if they have complementary strengths and therefore achieve a higher accuracy, precision, or recall when combined compared to when using each source individually. In our experimental evaluation (see section 5), we test the hypothesis 1 for experimental research question 1: *The combination of LLMs and a dictionary for value extraction leads to significantly higher accuracy compared to using either LLMs or a dictionary alone.* More specifically, we hypothesize that an LLMs-based value extraction can achieve high recall but lacks precision, whereas a dictionary-based approach can achieve high precision but lacks recall. Consequently, the combination of LLMs and a dictionary can achieve high precision and high recall. However, based on our results (see section 5.4), we reject the hypothesis 1. We observe that the combination of a dictionary and LLM achieves a lower accuracy than an LLM by itself. This implies that the combination of different value extraction sources does not necessarily increase the accuracy of the value extraction and might even decrease it.

Sub-research question 1 *How to identify context-specific values?* Section 1.5.2 mentions that value identification is a prerequisite for context-specific value extraction and therefore an answer to the main research question requires an answer to sub-research question 1. Contrary to many methods in the literature, VIVE separates value identification and value extraction into two steps and therefore includes a dedicated value identification module. While many methods are conceivable to perform value identification, VIVE constrains the options for its value identification module to hybrid intelligence methods that combine human and machine intelligence. For the reasons explained in 3.1, we consider a hybrid intelligence approach more suitable for value identification, than a completely automated method. In our instantiation of VIVE (section 4), we instantiate the value identification module based on Axies [39].

Independent of this study, the Netherlands Red Cross uses a list of context-specific topics to classify the messages from the Ukraine data set. The full list of context-specific topics can be found in Appendix C. Comparing the list of identified context-specific values (see section 5.3.1) to the list of context-specific topics illustrates that there is no clear distinction between personal values and topics. For example, the term *shelter* appears in both lists, therefore qualifying as both, a personal value and a topic. Section 1.5.1 defines a personal value as follows: *A personal value is a fundamental belief or principle that determines a person's attitudes, behaviors, and decision-making in life.* Following this definition, personal values can be viewed as a concept that overlaps with the concept of topics. However, we do not believe that personal values are a certain type of topic. For example, in this study, we identify the term *disappointment in this city/country* as a context-specific personal value of the Ukraine data set. However, none of the topics on the Red Cross's list of context-specific topics resembles it, and a Red Cross analyst stated that *disappointment in this city/country* is indeed not a topic. These findings imply that context-specific value identification cannot be substituted by topic modeling techniques.

Sub-research question 2 *How to represent personal values computationally?* Several different value representations are present in the reviewed literature and summarized in section 2.6. The VIVE pipeline includes a dedicated value representation module that transforms the identified context-specific personal values into a suitable format for value extraction. VIVE represents values as triples <*n, K, D*>, consisting of a name *n*, a list of keywords *K*, and a description *D* (see section 3.2). The conducted experiments show that the VIVE value representation principally allows a value extraction through LLM inference. We note that, when using LLM inference for value extraction, the value representation inherently influences the prompt strategy. That is because the formulation of prompts is constrained by the information present in the value representations, as can be seen in the two different contexts in section 4.4.1.

Our comparison of two value extraction classifiers using different value representations supports the hypothesis that the way personal values are represented impacts the performance of the value extraction. Our results show that representing a personal value not only through a name but also with keywords and a description significantly improves the accuracy of value extraction (see section 5.6). This indicates that a more extensive value representation leads to a higher accuracy, however, more work is needed to evaluate different value representations. Depending on the output of the value identification, additional features of the value representation are conceivable. For example, if known, the frequency *F* of a personal value in the data set can be included in the value representation, resulting in a quadruple <*n, K, D, F*>.

Sub-research question 3 *What is a function that indicates for any combination of a piece of text $d \in D_c$ and a context-specific value $v \in V_c$, whether v is referenced in d?* While the VIVE pipeline is designed to process a collection of texts (i.e. a natural language data set D_c, pertaining to a context *C*), it extracts values for each text $d \in D_c$ individually, once with an LLM-based algorithm (see algorithms 2 and 3) and once with a dictionary-based algorithm (see algorithms 4 and 5). When using an LLM-based algorithm for value extraction, a context-specific value $v \in V_c$ is considered referenced in a text $d \in D_c$, if the inference output of the LLM is affirmative to the question of whether *v* is referenced in *d*. When using a dictionary-based algorithm for value extraction, a context-specific value $v \in V_c$ is considered referenced in a text $d \in D_c$, if at least one keyword *k* that is stored in the dictionary in the list of keywords for *v*, occurs in *d*.

As described in section 3.3.4, VIVE combines the LLM-based and the dictionary-based algorithm to determine whether a value *v* is referenced in a text *d*. Therefore, the combination algorithms for multi-label (see algorithm 6) and single-label (see algorithm 7) value extraction are the answers that VIVE provides for sub-research question 3. In general terms, the way that a value extraction method answers sub-research question 3 depends on 1. the answers that the utilized value extraction sources provide and 2. the utilized combination algorithm.

In this study, we demonstrate how the individual outputs of multiple value extraction sources can be combined. Different ways of combining multiple value extraction sources can be more or less suitable, depending on the goal of the value extraction. For example, the VIVE value extraction algorithm (see algorithm 1) allows the choice of a *combination strategy*. Depending on the combination strategy, algorithm 6 optimizes the value extraction for recall or precision. Depending on the context, either optimizing for recall or precision can be the preferred strategy. For a value extraction method that utilizes more than two value extraction sources, more complex combination strategies are conceivable, for example, a weighing function that assigns a weight to each of the value extraction sources. In principle, a combination strategy should aim to combine "the best" of every value extraction source.

Meaning, that different value extraction sources might have different strengths (for example, high recall or high precision) and weaknesses, and the combination strategy should combine their strengths.

Sub-research question 4 *What is the accuracy, precision, recall, and usefulness of the proposed method for value extraction?* To answer sub-research question 4, we differentiate between multi-label value extraction (see definition 6) and single-label value extraction (see definition 7), and calculate the evaluation metrics separately, as specified in section 5.2.2.

The results of the experimental evaluation, as shown in sections 5.4, 5.5, and 5.6, show that LLMs can effectively extract personal values from natural language text. The results further show that a dictionary cannot effectively extract personal values from natural language text and a combination of a dictionary and LLMs is not preferable over the use of only LLMs. While principally capable of value extraction, our findings indicate that in terms of accuracy, LLMs have a performance ceiling of approximately 70 to 75 percent. However, when determining for a given value v and a given text d, whether v is in the set of referenced values V_d, LLMs appear to achieve an accuracy of around 90 percent.

This accuracy can be sufficient when using VIVE for summarization, as it provides a generally reliable overview of the data. For example, the participants of our user study (see section 5.7) indicate that the distribution of the personal values in the data set, as displayed in figures 5.19 and 5.20, represents useful information. On the other hand, 70 percent might not be an adequate level of accuracy when analyzing the referenced set of values V_d of a single text d from the data set.

Our results show that the obtained precision, recall, and F1 scores differ vastly between different personal values (see tables 5.8 and 5.9). For example, for the personal values *mental health* and *staying connected*, all LLM-based classifiers achieve an F1 score higher than 0.8, for both, multi-label and single-label value extraction. In contrast, for the personal value *disappointment in this city/country*, the highest obtained F1 score is 0.5 for the multi-label value extraction and 0.5333 for the single-label value extraction. This implies that some personal values are harder to extract than others. A possible reason is the degree of interpretability that the description of a personal value allows. We hypothesize that the more clear a description of a personal value is and the less room for interpretation it leaves, the more accurate LLMs can extract that value.

The accuracy of the value extraction likely also varies depending on the data set. In this study, we utilize the Ukraine data set, a data set of messages collected from Telegram groups and written by Ukrainian refugees or internally displaced people (see section 4.1). The messages were originally written in Ukrainian and translated to English by the Netherlands Red Cross. This presents a potential issue of the Ukraine data set for value extraction. Since personal values are often not explicitly mentioned, but rather referenced through nuances of a text, translating a text to a different language might lead to the loss of information relevant to the extraction of personal values. Furthermore, many messages in the Ukraine data set contain abbreviations of terms that are common to use for Ukrainians but likely unknown terms for an LLM. To examine how much the accuracy of value extraction varies depending on the data set, future work is required, more specifically, an instantiation of VIVE to at least one other data set.

6.2 Extensions and Future Work

There are several avenues for future work to expand this study and further investigate the task of value extraction. The following describes three such avenues that we deem particularly interesting.

6.2.1 Extension of VIVE

For the experiments conducted in this study, we instantiate VIVE to the context of the Ukraine data set, demonstrating how VIVE can be used to analyze feedback data from humanitarian projects. To do this, we essentially perform one forward pass through the VIVE pipeline, as displayed in figure 4.1. Meaning, we perform value identification and value extraction once for the Ukraine data set. However, in principle, VIVE can be extended to a continuous loop. Figure 6.1 illustrates a continuous loop as a possible extension of VIVE. Following the value extraction module, a value reporting module can summarize and further analyze the labeled data sets. The value reporting module can include various summarization techniques to give the user an overview of the labeled data set that is the output of the value extraction module. For our user study, described in section 5.7, we essentially perform value reporting, by displaying summary statistics of the value extraction, such as charts. As figure 6.1 illustrates, the user can interact with the value reporting module and request information. Examples of a concrete realization of the value reporting module are a chatbot or an interactive dashboard. Value reporting further addresses requirement 1 from our problem investigation, to automatically extract relevant information and potentially addresses requirement 3, to translate the information into actionable output. An interactive value reporting module also addresses the feedback of a participant from our user study that an interactive output of VIVE would be useful.

VIVE becomes a loop because, at any point in time, the user has the option to interact with the value identification module and the value reporting module. Any interaction with the value identification module impacts the output of the value extraction module and therefore the value reporting. The other way around, the output of the value reporting module can lead the user to redo or continue the value identification. As opposed to the experiments conducted for this study, in which we use a fixed, pre-determined data set, it is conceivable to use VIVE with live data. Data can be fetched via server endpoints or in discrete time steps.

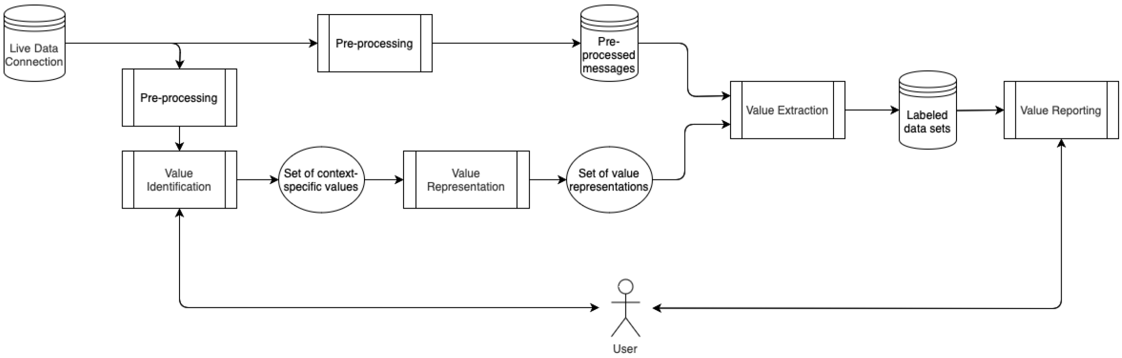


FIGURE 6.1: Possible extension of the VIVE pipeline to a continuous loop.

6.2.2 Annotator-centric Value Extraction

To evaluate our instantiation of VIVE, two Red Cross analysts annotated a subset of the Ukraine data set with five of the context-specific personal values (see section 5.2.1). Despite pre-selecting messages for the annotation, the two annotators disagreed on approximately 14 percent of the annotated messages, and the resulting intercoder reliability, measured by the Cohen Kappa (CK) is 0.8459. While this is considered a high intercoder reliability [42], it supports the assumption of Sagiv et al. [61] that the exact understanding of personal values is unique for every individual. Put in a broader context, it is widely acknowledged that natural language understanding tasks, such as value extraction, contain a degree of subjectivity. Fornciari et al. [20] identify this as a fundamental constraint of supervised learning that uses human-annotated labels. Supervised learning considers the human-annotated labels as the ground truth. However, two different annotators might not even agree on the correct annotation. Consequently, we hypothesize that to further increase the accuracy of the value extraction, the extraction needs to be personalized to the individual annotator.

In their article "Annotator-Centric Active Learning for Subjective NLP Tasks", Meer et al. [43] propose ACAL, an active learning approach that accounts for different opinions amongst annotators due to the subjectivity of an annotation task. As figure 6.2 shows, as opposed to traditional active learning, ACAL contains an annotator selection strategy. For each selected sample, the annotator selection strategy selects an annotator. The goal is to approximate the distribution of human judgment for a given sample.

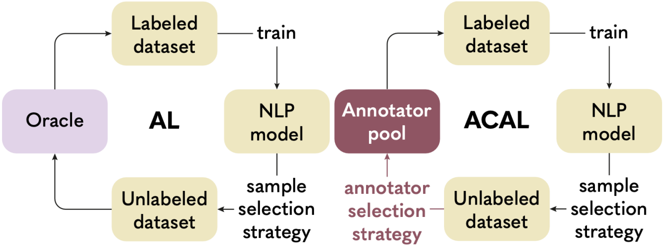


FIGURE 6.2: Annotator-centric active learning (taken from [43]).

We envision annotator-centric value extraction as a value extraction method (and potentially part of the VIVE pipeline) that learns the individual value definitions and value taxonomy of the user. To evaluate such a method, annotator-centric evaluation metrics are required.

6.2.3 Prompt Strategies

Section 3.3.2 briefly explains the impact of the prompt strategy on the inference output when using LLMs for value extraction. It is conceivable that we can significantly improve the performance (accuracy, precision, recall, and F1-score) of the value extraction by further developing our prompt strategies. For example, one option is to include more information in the context that is given to the LLM. The two contexts used in this study (see section 4.4.1) include a short description of a role and the identified context-specific personal values. However, they do not include any example texts with their referenced values or an explanation of what personal values are. This is because depending on the chosen method for value identification, no example texts with their referenced values might be available. For example, Axies, the chosen value identification method in this study's instantiation of VIVE, solely outputs a list of value representations that do not contain example messages.

Besides the performance, the chosen prompt strategy significantly impacts the amount of computational resources needed for value extraction. For example, in this study, the multi-label prompt strategy consists of one prompt per context-specific value. Assuming a constant runtime per prompt, the time needed to extract the referenced values V_d for a text d increases linearly with the number of context-specific values. In contrast, the below prompt shows an example of how to extract the referenced values V_d for a text d with a single prompt. In this case, the time needed stays constant for any number of context-specific values.

Alternative multi-label prompt template:
"Which of the following personal values are underlying values for the following message?
Message: <MESSAGE>
Personal Values: <VALUE NAME FOR EACH VALUE>
To answer the question, consider for each personal value, whether the following sentence is a correct statement: The author composed this message, because <VALUE NAME> is important to him/her?
Answer only by listing the underlying values in this format
[<value1>,<value2>,...,<valueN>]"

Chapter 7

Conclusion

In the context of AI-supported decision-making, particularly within humanitarian organizations, ensuring value alignment is crucial to protect the rights and well-being of vulnerable people. In this study, we propose and evaluate VIVE (*Value Identification and Value Extraction*), an end-to-end method for identifying and extracting context-specific personal values from natural language text. VIVE consists of several modules that form a pipeline to analyze the context-specific personal values that are referenced in a data set of natural language texts. The first module in the pipeline is the value identification module. It combines human and artificial intelligence to identify a set of context-specific personal values. The second module is the value representation module, formatting the set of identified, context-specific values in a way that makes it suitable for value extraction. The third module of the VIVE pipeline is the value extraction module that uses the value representations to extract references to context-specific personal values from the data set of natural language texts.

This study provides two main contributions to the field of natural language processing (NLP). The first main contribution is the proposal of a novel end-to-end method for the identification and extraction of context-specific personal values. The second main contribution is the proposal of a method for value extraction based on large language modules. We find that LLMs exhibit a level of natural language understanding that allows them to sufficiently comprehend the nuances and subtleties of human language to extract personal values. Consequently, our answer to the main research question of this study - *How to automatically extract context-specific personal values from natural language text?* - is that personal values can be automatically extracted from natural language text through LLM inference.

To evaluate the value extraction capabilities of VIVE, we conduct three experiments: First, we examine the accuracy, precision, recall, and F1-score of the value extraction module when using only a dictionary, only an LLM, or a combination of a dictionary and an LLM. We show through statistical testing that using only an LLM yields the best accuracy for the task of value extraction. Second, we compare three state-of-the-art LLMs to examine the impact of using different LLMs. We observe that using different LLMs does not significantly impact the accuracy of the value extraction. Third, we examine the impact of the value representation on the value extraction. We observe that the way personal values are represented matters insofar as a less extensive value representation leads to a significantly worse accuracy. The results of our user study show that VIVE addresses the requirements of humanitarian organizations with regard to processing feedback data from humanitarian programs.

As directions for future work, we propose an extension of the VIVE pipeline to a continuous loop that includes a value reporting module. To increase the accuracy of the value extraction, we suggest exploring annotator-centric value extraction and further developing the prompt strategies.

Bibliography

[1] Sallam Abualhaija et al. "Replication in Requirements Engineering: the NLP for RE Case". In: *ACM Transactions on Software Engineering and Methodology* (2024).

[2] Luigi Asprino et al. "Uncovering values: Detecting latent moral content from natural language with explainable and non-trained methods". In: *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Association for Computational Linguistics. 2022, pp. 33–41.

[3] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. "A survey on context-aware systems". In: *International Journal of ad Hoc and ubiquitous Computing* 2.4 (2007), pp. 263–277.

[4] Sugato Basu, Arindam Banerjee, and Raymond J Mooney. "Active semi-supervision for pairwise constrained clustering". In: *Proceedings of the 2004 SIAM international conference on data mining*. SIAM. 2004, pp. 333–344.

[5] Ana Beduschi. "Harnessing the potential of artificial intelligence for humanitarian action: Opportunities and risks". In: *International Review of the Red Cross* 104.919 (2022), pp. 1149–1169.

[6] Gregory Biegel and Vinny Cahill. "A framework for developing mobile, context-aware applications". In: *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the*. IEEE. 2004, pp. 361–365.

[7] Steven Bird. "NLTK: the natural language toolkit". In: *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. 2006, pp. 69–72.

[8] A van den Bosch et al. "Artificial Intelligence Research Agenda for the Netherlands". In: ().

[9] RL Boyd. *Meaning extraction helper (2.1. 07)*. 2018.

[10] Ryan Boyd et al. "Values in words: Using language to evaluate and understand personal values". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 9. 1. 2015, pp. 31–40.

[11] Rosanna L Breen. "A practical guide to focus-group research". In: *Journal of geography in higher education* 30.3 (2006), pp. 463–475.

[12] Yung-Chun Chang, Chih-Hao Ku, and Chun-Hung Chen. "Social media analytics: Extracting and visualizing Hilton hotel ratings and reviews from TripAdvisor". In: *International Journal of Information Management* 48 (2019), pp. 263–279.

[13] Brian Christian. *The alignment problem: Machine learning and human values*. WW Norton & Company, 2020.

[14] Cindy K Chung and James W Pennebaker. "Revealing dimensions of thinking in open-ended self-descriptions: An automated meaning extraction method for natural language". In: *Journal of research in personality* 42.1 (2008), pp. 96–132.

[15] Davide Dell'Anna, Fatma Başak Aydemir, and Fabiano Dalpiaz. "Evaluating classifiers in SE research: the ECSER pipeline and two replication studies". In: *Empirical Software Engineering* 28.1 (2023), p. 3.

[16] Janez Demšar. "Statistical comparisons of classifiers over multiple data sets". In: *The Journal of Machine learning research* 7 (2006), pp. 1–30.

[17] Anind K Dey. "Understanding and using context". In: *Personal and ubiquitous computing* 5 (2001), pp. 4–7.

[18] Ethan Fast, Binbin Chen, and Michael S Bernstein. "Empath: Understanding topic signals in large-scale text". In: *Proceedings of the 2016 CHI conference on human factors in computing systems*. 2016, pp. 4647–4657.

[19] Luis Fernandez-Luque and Muhammad Imran. "Humanitarian health computing using artificial intelligence and social media: A narrative literature review". In: *International journal of medical informatics* 114 (2018), pp. 136–142.

[20] Tommaso Fornaciari et al. "Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. 2021.

[21] Batya Friedman. *Human values and the design of computer technology*. 72. Cambridge University Press, 1997.

[22] Batya Friedman et al. "Value sensitive design and information systems". In: *Early engagement and new technologies: Opening up the laboratory* (2013), pp. 55–95.

[23] Erich Gamma et al. *Design patterns: elements of reusable object-oriented software*. Pearson Deutschland GmbH, 1995.

[24] Sahar Ghannay et al. "Word embedding evaluation and combination". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2016, pp. 300–305.

[25] Shantanu Godbole and Sunita Sarawagi. "Discriminative methods for multilabeled classification". In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2004, pp. 22–30.

[26] Jesse Graham et al. "Moral foundations theory: The pragmatic validity of moral pluralism". In: *Advances in experimental social psychology*. Vol. 47. Elsevier, 2013, pp. 55–130.

[27] Tao Gu et al. "An ontology-based context model in intelligent environments". In: *arXiv preprint arXiv:2003.05055* (2020).

[28] Jacqueline M Guarte and Erniel B Barrios. "Estimation under purposive sampling". In: *Communications in Statistics-Simulation and Computation* 35.2 (2006), pp. 277–284.

[29] Samaneh Heidari, Maarten Jensen, and Frank Dignum. "Simulations with values". In: *Advances in Social Simulation: Looking in the Mirror*. Springer. 2020, pp. 201–215.

[30] Steffen Herbold. "Autorank: A python package for automated ranking of classifiers". In: *Journal of Open Source Software* 5.48 (2020), p. 2173.

[31] Frederic R Hopp et al. "The extended Moral Foundations Dictionary (eMFD): Development and applications of a crowd-sourced approach to extracting moral intuitions from text". In: *Behavior research methods* 53 (2021), pp. 232–246.

[32] Frankie James. "Modified kneser-ney smoothing of n-gram models". In: *Research Institute for Advanced Computer Science, Tech. Rep. 00.07* (2000).

[33] Albert Q Jiang et al. "Mistral 7B". In: *arXiv preprint arXiv:2310.06825* (2023).

[34] Ankur Joshi et al. "Likert scale: Explored and explained". In: *British journal of applied science & technology* 7.4 (2015), pp. 396–403.

[35] Ashraf M Kibriya et al. "Multinomial naive bayes for text categorization revisited". In: *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17*. Springer. 2005, pp. 488–499.

[36] Sidney I Landau. *Cambridge Dictionary of American English*. Cambridge University Press, 1999.

[37] Shiyao Li et al. "Evaluating Quantized Large Language Models". In: *arXiv preprint arXiv:2402.18158* (2024).

[38] Enrico Liscio et al. "A Collaborative Platform for Identifying Context-Specific Values". In: (2021).

[39] Enrico Liscio et al. "Axies: Identifying and Evaluating Context-Specific Values." In: *AAMAS*. 2021, pp. 799–808.

[40] Xiao Liu et al. "GPT understands, too". In: *AI Open* (2023).

[41] David M Markowitz. "The meaning extraction method: An approach to evaluate content patterns from large-scale language data". In: *Frontiers in Communication* 6 (2021), p. 588823.

[42] Mary L McHugh. "Interrater reliability: the kappa statistic". In: *Biochemia medica* 22.3 (2012), pp. 276–282.

[43] Michiel van der Meer et al. "Annotator-Centric Active Learning for Subjective NLP Tasks". In: *arXiv preprint arXiv:2404.15720* (2024).

[44] Quim Motger et al. "T-FREX: A Transformer-based Feature Extraction Method from Mobile App Reviews". In: *arXiv preprint arXiv:2401.03833* (2024).

[45] Kamal Nigam, John Lafferty, and Andrew McCallum. "Using maximum entropy for text classification". In: *IJCAI-99 workshop on machine learning for information filtering*. Vol. 1. 1. Stockholom, Sweden. 1999, pp. 61–67.

[46] Nardine Osman and Mark d'Inverno. "A computational framework of human values for ethical AI". In: *arXiv preprint arXiv:2305.02748* (2023).

[47] James W Pennebaker, Martha E Francis, and Roger J Booth. "Linguistic inquiry and word count: LIWC 2001". In: *Mahway: Lawrence Erlbaum Associates* 71.2001 (2001), p. 2001.

[48] Ibo Van de Poel. "Design for value change". In: *Ethics and Information Technology* 23.1 (2021), pp. 27–31.

[49] Alina Pommeranz et al. "Elicitation of situated values: need for tools to help stakeholders and designers to reflect and communicate". In: *Ethics and Information Technology* 14.4 (2012), pp. 285–303.

[50] Vladimir Ponizovskiy et al. "Development and validation of the personal values dictionary: A theory–driven tool for investigating references to basic human values in text". In: *European Journal of Personality* 34.5 (2020), pp. 885–902.

[51] Wisam A Qader, Musa M Ameen, and Bilal I Ahmed. "An overview of bag of words; importance, implementation, applications, and challenges". In: *2019 international engineering conference (IEC)*. IEEE. 2019, pp. 200–204.

[52] Alec Radford et al. “Improving language understanding by generative pre-training”. In: (2018).

[53] Radim Řehřek, Petr Sojka, et al. “Gensim—statistical semantics in python”. In: *Retrieved from genism. org* (2011).

[54] Nils Reimers and Iryna Gurevych. “Sentence-bert: Sentence embeddings using siamese bert-networks”. In: *arXiv preprint arXiv:1908.10084* (2019).

[55] Sonia Roccas and Lilach Sagiv. “Personal values and behavior: Taking the cultural context into account”. In: *Social and Personality Psychology Compass* 4.1 (2010), pp. 30–41.

[56] Milton Rokeach. “Rokeach value survey”. In: *The nature of human values.* (1967).

[57] Milton Rokeach. *The nature of human values.* Free press, 1973.

[58] Robert Rosenthal, Harris Cooper, Larry Hedges, et al. “Parametric measures of effect size”. In: *The handbook of research synthesis* 621.2 (1994), pp. 231–244.

[59] Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

[60] Stuart Russell, Daniel Dewey, and Max Tegmark. “Research priorities for robust and beneficial artificial intelligence”. In: *AI magazine* 36.4 (2015), pp. 105–114.

[61] Lilach Sagiv et al. “Personal values in human life”. In: *Nature human behaviour* 1.9 (2017), pp. 630–639.

[62] Zeeshan Saleem et al. “Context-aware text classification system to improve the quality of text: A detailed investigation and techniques”. In: *Concurrency and Computation: Practice and Experience* 35.15 (2023), e6489.

[63] Shalom H Schwartz. “An overview of the Schwartz theory of basic values”. In: *Online readings in Psychology and Culture* 2.1 (2012), p. 11.

[64] Shalom H Schwartz. “Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries”. In: *Advances in experimental social psychology*. Vol. 25. Elsevier, 1992, pp. 1–65.

[65] Kyarash Shahriari and Mana Shahriari. “IEEE standard review—Ethically aligned design: A vision for prioritizing human wellbeing with artificial intelligence and autonomous systems”. In: *2017 IEEE Canada International Humanitarian Technology Conference (IHTC)*. IEEE. 2017, pp. 197–201.

[66] Mohammad S Sorower. “A literature survey on algorithms for multi-label learning”. In: *Oregon State University, Corvallis* 18.1 (2010), p. 25.

[67] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. “The general inquirer: A computer approach to content analysis.” In: (1966).

[68] Thomas Strang and Claudia Linnhoff-Popien. “A context modeling survey”. In: *Workshop Proceedings*. 2004.

[69] Gemma Team et al. “Gemma: Open models based on gemini research and technology”. In: *arXiv preprint arXiv:2403.08295* (2024).

[70] Livia Teernstra et al. “The morality machine: Tracking moral values in tweets”. In: *Advances in Intelligent Data Analysis XV: 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings 15*. Springer. 2016, pp. 26–37.

[71] Heidi Vainio-Pekka et al. "The Role of Explainable AI in the Research Field of AI Ethics". In: *ACM Transactions on Interactive Intelligent Systems* (2023).

[72] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[73] Alex Wang et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding". In: *arXiv preprint arXiv:1804.07461* (2018).

[74] Jacques de Wet, Daniela Wetzelhütter, and Johann Bacher. "Revisiting the transsituationality of values in Schwartz's Portrait Values Questionnaire". In: *Quality & Quantity* 53 (2019), pp. 685–711.

[75] Roel J Wieringa. *Design science methodology for information systems and software engineering*. Springer, 2014.

[76] Eva M Witesman and Lawrence C Walters. "Modeling public decision preferences using context-specific value hierarchies". In: *The American Review of Public Administration* 45.1 (2015), pp. 86–105.

[77] Shakti Kumar Yadav et al. "Sampling methods". In: *Biomedical Statistics: A Beginner's Guide* (2019), pp. 71–83.

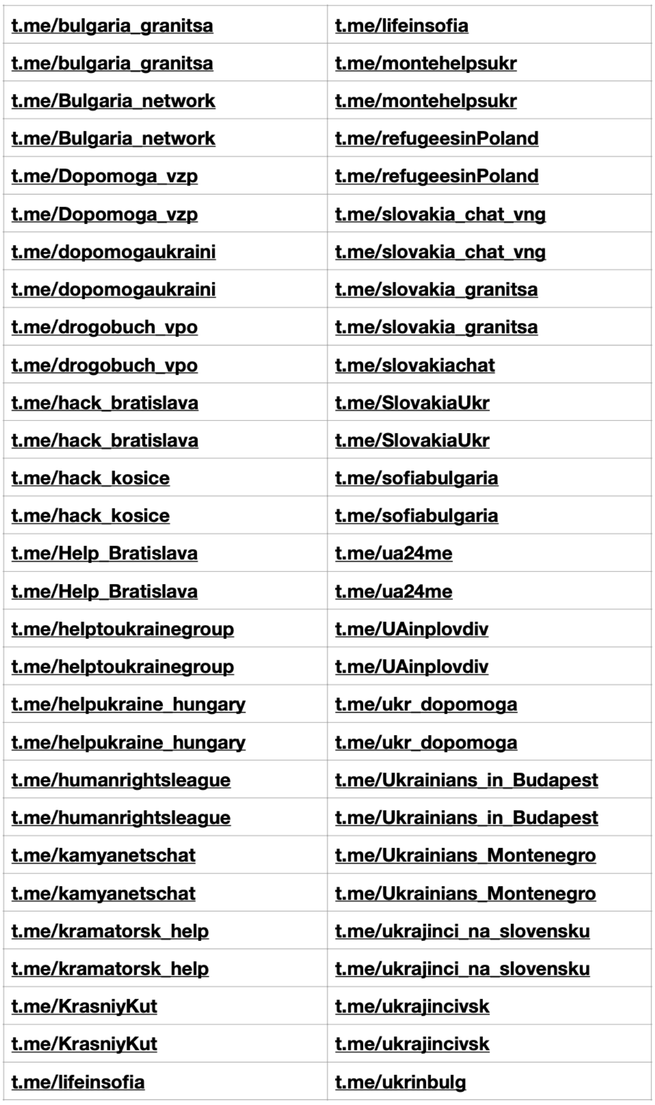
[78] Takayuki Yamaguchi, Shunichi Hattori, and Yasufumi Takama. "Proposal of personal-value-based item modeling and its application to explanation of recommendation". In: *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. IEEE. 2015, pp. 58–63.

[79] Zixuan Zhou et al. "A Survey on Efficient Inference for Large Language Models". In: *arXiv preprint arXiv:2404.14294* (2024).

Appendix A

Telegram Group Names

Below is a list of all Telegram[1] groups from which the messages in the Ukraine data set were collected.



t.me/bulgaria_granitsa	t.me/lifeinsofia
t.me/bulgaria_granitsa	t.me/montehelpsukr
t.me/Bulgaria_network	t.me/montehelpsukr
t.me/Bulgaria_network	t.me/refugeesinPoland
t.me/Dopomoga_vzp	t.me/refugeesinPoland
t.me/Dopomoga_vzp	t.me/slovakia_chat_vng
t.me/dopomogaukraini	t.me/slovakia_chat_vng
t.me/dopomogaukraini	t.me/slovakia_granitsa
t.me/drogobuch_vpo	t.me/slovakia_granitsa
t.me/drogobuch_vpo	t.me/slovakiachat
t.me/hack_bratislava	t.me/SlovakiaUkr
t.me/hack_bratislava	t.me/SlovakiaUkr
t.me/hack_kosice	t.me/sofiabulgaria
t.me/hack_kosice	t.me/sofiabulgaria
t.me/Help_Bratislava	t.me/ua24me
t.me/Help_Bratislava	t.me/ua24me
t.me/helptoukrainegroup	t.me/UAinplovdiv
t.me/helptoukrainegroup	t.me/UAinplovdiv
t.me/helpukraine_hungary	t.me/ukr_dopomoga
t.me/helpukraine_hungary	t.me/ukr_dopomoga
t.me/humanrightsleague	t.me/Ukrainians_in_Budapest
t.me/humanrightsleague	t.me/Ukrainians_in_Budapest
t.me/kamyanetschat	t.me/Ukrainians_Montenegro
t.me/kamyanetschat	t.me/Ukrainians_Montenegro
t.me/kramatorsk_help	t.me/ukrajinci_na_slovensku
t.me/kramatorsk_help	t.me/ukrajinci_na_slovensku
t.me/KrasniyKut	t.me/ukrajincivsk
t.me/KrasniyKut	t.me/ukrajincivsk
t.me/lifeinsofia	t.me/ukrinbulg

[1] Telegram is a free instant messaging service:
`https://web.telegram.org/a/`
`https://en.wikipedia.org/wiki/Telegram_(software)`

Appendix B

Identified Values for the Ukraine data set

The table B.1 on the next page presents a full representation of all 16 context-specific personal values that were identified for the context of the Ukraine data set. The table is referenced in section 4.3.

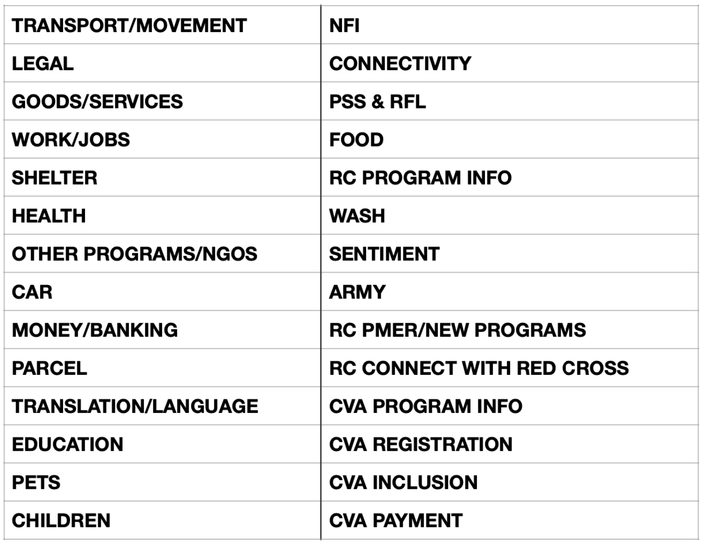
TABLE B.1: Context-specific personal values of the Ukraine data set

Name	**Keywords**	**Description**
shelter	housing	Ukrainian refugees need a shelter/place to stay or to live when they flee to other regions of Ukraine or to other/neighbouring countries. This is of paramount importance.
important parcel	package, parcel, documents	They look for carriers to send a document or a medicine or something important to Ukraine or to other countries where their relatives live.
mental health	psychological support, psychological health	They look for and offer psychological support because many people are traumatized by war.
financial matters	money, banking	Financial matters are important for them, as they are now in a new country/city and need money for everything.
staying connected	mobile phone, connectivity, Internet	Good and inexpensive service provider (mobile phone, Internet) is very important to stay in touch with their families/husbands.
staying warm in winter	household items	Self-explanatory.
work	job, salary	Most of them look for jobs in a new country/city to be financially independent. This is their mentality.
getting around	ticket, public transport	As they are in a new country/city, they have many questions on how to get around.
obeying traffic rules	car, car documents	Many travel by car and need to know things about traffic rules and necessary documents.
health	health check up, doctor	Many have health problems (many pregnant women or women with little children need doctors' help).
pet	bring a pet across the border, vet	Many people affected bring their pets to the new countries. They flee the war and also want to save their pets.
everyday life	service, goods, repair specialist	When they settle a bit, they need different goods and services.
translation service	translator, language course	They moved to a new country, so they do not speak the language, thus we see a lof of translation and language courses requests.
important documents	ID, Passport, Guardianship, Lawyer, legal	People need to have new documents done in a new country. They need lawyer and notary services.
disappointment in this city/country	go back home	From time to time we see messages when people get disappointed in a new place and decide to go back to Ukraine even though it is unsafe over there.
help for refugees	humanitarian aid, Red Cross	Humanitarian and other kinds of help is needed for the people affected.

Appendix C

Ukraine data set Topics

Below is a list of context-specific topics for the Ukraine data set, created by analysts of the Netherlands Red Cross. The list is briefly discussed in section 6.1 of the discussion.



TRANSPORT/MOVEMENT	NFI
LEGAL	CONNECTIVITY
GOODS/SERVICES	PSS & RFL
WORK/JOBS	FOOD
SHELTER	RC PROGRAM INFO
HEALTH	WASH
OTHER PROGRAMS/NGOS	SENTIMENT
CAR	ARMY
MONEY/BANKING	RC PMER/NEW PROGRAMS
PARCEL	RC CONNECT WITH RED CROSS
TRANSLATION/LANGUAGE	CVA PROGRAM INFO
EDUCATION	CVA REGISTRATION
PETS	CVA INCLUSION
CHILDREN	CVA PAYMENT