JVLC

Journal of Visual Language and Computing

Volume 2023, Number 2

DOI: 10.18293/JVLC2023-N2

Journal preparation, editing and printing are sponsored by KSI Research Inc.

Journal of
Visual Language and Computing

An Open Access Journal published by

KSI Research Inc.

156 Park Square Lane, Pittsburgh, PA 15238 USA

JVLC Editorial Board

Journal of
Visual Language and Computing

Volume 2023, Number 2

December 2023

Table of Contents

Regular Papers

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Unleashing the Power of NLP Models for Semantic Consistency Checking of Automation Rules

Bernardo Breve[a], Gaetano Cimino[a], Vincenzo Deufemia[a] and Annunziata Elefante[a,*]

[a]University of Salerno, via Giovanni Paolo II, Fisciano (SA), 84084, Italy

ARTICLE INFO

Article History:
Submitted 4.30.2023
Revised 7.31.2023
Accepted 8.1.2023

Keywords:
Trigger-action rules
Semantic consistency checking
NLP
IoT platforms

ABSTRACT

Trigger-Action Platforms (TAPs) empower users to automate tasks involving smart devices, allowing them to either create rules from scratch or access a catalog of existing user-defined rules. Users can explore the catalog and find rules based on their interests, relying on the so-called User-defined descriptions (UDDs) provided by the rules' creators. However, TAPs lack a mechanism to verify or regulate these descriptions, resulting in potential inaccuracies or errors. This poses challenges for users when seeking relevant rules, as descriptions may present misleading or irrelevant information.

In this paper, we propose a novel approach to semantically validate the consistency between a rule's UDD and its trigger-action components. To accomplish this objective, we used rules derived from a widely used TAP, i.e., If-This-Then-That (IFTTT). From the automation rules, we constructed a dataset of 20,000 samples, and we assigned them labels representing four distinct classes of semantic consistency. For two of these classes, we leveraged the capabilities of a Large Language Model (LLM) to edit the user descriptions, significantly reducing manual effort while ensuring coherent samples. In order to evaluate the semantic consistency, we employed three NLP-based classification models, fine-tuned on the dataset we created. This allowed us to assess the effectiveness of our proposed approach. Among the models, the BERT-based model demonstrated superior performance, achieving an accuracy value of 99%.

1. Introduction

The Internet of Things (IoT) has revolutionized various industries and aspects of daily life by connecting physical devices and enabling data exchange through sensors and network connectivity [20]. Intelligent IoT systems and devices enable the automation of tasks and efficient data management, leading to the emergence of "smart devices" that enhance user experiences. Trigger-Action Platforms (TAPs) [13, 34] are crucial pieces of software in IoT systems, as they allow users to create automation rules that trigger specific actions based on conditions, such as turning on the light automatically at a certain time. TAPs are particularly valuable in End-User Development (EUD) [19, 26], as they empower users to define their automation tasks without the need for extensive programming knowledge in a very simple and intuitive way. This user-friendly approach opens up endless possibilities for customization and tailoring automation to individual needs and preferences.

Each rule is composed of a trigger component, defining the event that activates the rule, and an action component, detailing the operation to be executed to achieve the desired behavior. Additionally, TAPs often allow users to provide rule-specific information in the form of a textual description, known as the *User-defined description* (UDD), which succinctly summarizes the rule's behavior.

The If-This-Then-That (IFTTT) [1] platform serves as the primary and most widely used TAP in the market. Since its inception in 2010, the platform has garnered a substantial

This work has been supported by the Italian Ministry of University and Research (MUR) under grant PRIN 2017 "EMPATHY: Empowering People in deAling with internet of THings ecosYstems" (Progetti di Rilevante Interesse Nazionale – Bando 2017, Grant 2017MX9T7H).

*Corresponding author
bbreve@unisa.it (B. Breve); gcimino@unisa.it (G. Cimino); deufemia@unisa.it (V. Deufemia); anelefante@unisa.it (A. Elefante)
ORCID(s): 0000-0002-3898-7512 (B. Breve); 0000-0001-8061-7104 (G. Cimino); 0000-0002-6711-3590 (V. Deufemia); 0009-0001-7141-6105 (A. Elefante)

[1] https://ifttt.com

DOI reference number: 10.18293/JVLC2023-N2-035

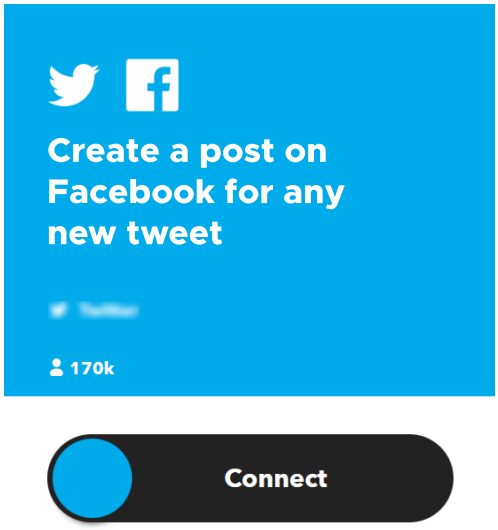


Figure 1: An rule's example with the associated UDD

and ever-growing community of followers. A notable advantage of IFTTT is its extensive catalog of rules, known as *applets*, shared by community members. In this context, UDDs play a crucial role in helping users comprehend rule behaviors while browsing the catalog. Figure 1 presents an example of an IFTTT rule from the catalog (with the author's name blurred for privacy), with the rule's behavior summarized through its UDD. In this specific instance, the rule automates the synchronization of any new tweet published by a user on Twitter to his/her personal Facebook Page.

Sadly, the utilization of TAPs and automation rules introduces security and privacy risks [10, 16, 38], as they may grant access to sensitive data and be misinterpreted in their behavior. The presence of IoT devices also poses potential risks, as they could be exploited by malicious individuals for cyber attacks [1, 25]. Additionally, users' interactions with TAPs can inadvertently introduce cybersecurity threats [33]. The creation of rules through TAPs may carry inherent risk, particularly due to the average user's level of technical knowledge, which may not be sufficient to fully comprehend the potential consequences of seemingly innocuous rules [11, 30]. For instance, a rule like "If the last family member leaves the house, then turn off the lights" could inadvertently disclose valuable information to malicious individuals, providing insights into when the user's home will be empty. To address these challenges, researchers have proposed tailor-made solutions to safeguard users' privacy and security within intelligent environments [3, 9, 35].

The existence of fields like UDDs also raises significant concerns for users [2, 7], yet this aspect has received limited attention in the literature. TAPs such as IFTTT lack active control over the content authors may input as UDDs, granting them the freedom to write anything to describe the behavior of their rules. This unrestricted approach may give rise to several issues. Firstly, rule creators may enter UDDs that are completely unrelated not only to the rule's behavior but also fail to conform to the typical characteristics of a description, such as "10 Things You Need To Know!". Consequently, such rules become virtually impossible for users to discover. Secondly, rules with imprecise UDDs might surface in search results for other types of rules, making it even more challenging for users to find rules that suit their requirements accurately. Moreover, poor UDDs may lead to a lack of understanding of the rules' intended behavior, as the UDD serves as a showcase for the rule's purpose. Finally, TAPs with shared rule catalogs can potentially expose users to risks if malicious authors hide harmful behaviors behind misleading descriptions. For instance, consider the rule's UDD in Figure 1, but assume that its trigger and action components instead are "If anyone in your area publishes a new tweet" and "then create a post on Facebook", respectively. The combination of these components could potentially result in the malicious posting of embarrassing and unwanted texts. As a consequence, such information may automatically be published on Facebook and shared with an even wider audience without the user's consent or awareness, unlike what the UDD might suggest. Since both the UDD and the actual trigger-action components involve the same services, i.e., *Twitter* and *Facebook*, users might be deceived into activating such a rule, unaware of the potential harm. These concerns emphasize the need for attention and possible solutions in this domain.

To mitigate this risk is crucial to preserve the semantic consistency between a rule's behavior and its UDD. This should be done by means of approaches that analyze UDDs to identify potential misalignments, safeguarding users from potential threats that may arise due to deceptive or misleading rule descriptions. Furthermore, maintaining semantic consistency between a rule's behavior and its UDD is important also for those approaches relying on the analysis of UDDs to identify potential user privacy or security-related harm caused by rules [5, 15, 31].

In response to the mentioned concerns, this paper proposes a novel approach that addresses the identified issues effectively. In previous work [6], we proposed a Bidirectional Encoder Representations from Transformers (BERT)-based model evaluating the semantic consistency between UDDs and the trigger-action components of rules according to two consistency classes, i.e., either complete consistency between a UDD and its trigger-action component, or complete inconsistency. In this paper, we further extend our methodology by considering two additional classes of semantic consistency, i.e., trigger-side inconsistent only and action-side inconsistent only. Furthermore, we also compared the BERT-based model [14] with two other transformer-based NLP ones: the Generative Pre-trained Transformer 2 (GPT-2) model [28], which is a decoder-only model, and the Text-To-Text Transfer Transformer (T5) model [29], which combines both encoder and decoder components.

To obtain the samples to be considered for the training following the new distribution of classes, we constructed a dataset encompassing all conceivable scenarios related to the generation and sharing of UDDs associated with automation rules. Leveraging a Large Language Model (LLM) [24],

we significantly reduce manual efforts while ensuring dataset heterogeneity. The resulting dataset contains 20,000 samples, where the actual behavior of each rule is represented by a textual pattern derived from its components. These patterns, along with the rule's UDDs, serve as inputs for the classification models, which calculate a semantic similarity score between the two texts. The results demonstrated the effectiveness of these models in categorizing semantic consistency, achieving an overall accuracy rate of approximately 99%, 97%, and 91% respectively.

The paper is organized as follows: Section 2 discusses the state of the art on semantic analysis of automation rules. Section 3 presents an overview of the overall methodology, detailing the model's general architecture and outlining the dataset construction process. Moving on to Section 4, we describe the process of generating the dataset, encompassing all possible types of UDDs that a user might define. For the task of checking the semantic consistency between a rule's UDD and its trigger-action components, Section 5 provides an in-depth explanation of the NLP classification models employed. In Section 6, we present the results of the experimental evaluation, assessing the effectiveness of the classification models. Finally, Section 7 concludes the manuscript and provides future directions for our proposal.

2. Related Work

This section presents an overview of the main research endeavors in the world of semantic analysis of trigger-action rules. Previous studies in the literature have primarily concentrated on language-to-code approaches, extracting executable code from rule descriptions. Alternatively, there have been efforts to improve user experience by developing advanced graphical interfaces or employing sequence-to-sequence models for the automatic generation of rule components, streamlining the rule creation process for users.

Utilizing natural language to program computers has the potential to enhance accessibility to modern technology, especially for inexperienced users [22]. One approach to achieve this is through the development of language-to-code translators, which aid in creating trigger-action rules tailored to user needs. By employing a semantic parser, natural language descriptions can be converted into executable code, streamlining the process of rule customization and making it more user-friendly for a broader audience. In [27], Quirk *et al.* designed a language-to-code approach for natural language programming. They collected a significant number of rule-description pairs from the IFTTT website and used them to train semantic parser learners capable of effectively interpreting natural language descriptions and mapping them to executable code. The IF-THEN statements were represented using Abstract Syntax Trees (ASTs), with each node denoting a specific text construct and capturing its structural and content-related details. The constructed ASTs were then fed to several classifiers, which iteratively searched for the most likely derivation, refining the training data to achieve desired performance. Another study [21] by Chen *et al.* proposed a neural network architecture for automatically translating natural language descriptions into IF-THEN rules. They introduced an attention mechanism called *Latent Attention*, which computed the importance of each word in the description to predict rule components in a two-stage process. Additionally, Yusuf *et al.* presented *RecipeGen*, a deep learning-based approach that utilizes a Transformer sequence-to-sequence architecture to generate IF-THEN rules from natural language descriptions [37]. This model treated the problem as a sequence learning and generation task, effectively capturing implicit relations between rule components. To enhance generation performance, RecipeGen relied on autoencoding pre-trained models to initialize the encoder's parameters in the sequence-to-sequence model.

Prior studies have mainly focused on interactions that involve a user's request and the system's response in the form of interpretation. However, it is essential to engage the user in an interactive dialogue to validate and refine their intentions, leading to the creation of complete and accurate rules. Addressing this aspect, Corno *et al.* proposed *HeyTAP* [12], a conversational and semantic-powered platform that can map abstract user needs to executable IF-THEN rules. HeyTAP utilizes a multimodal interface to interact with the user and extract personalization intentions for various contexts. An exploratory experiment involving 8 users demonstrated HeyTAP's effectiveness in guiding participants from abstract needs to concrete IF-THEN rules, which can be executed by contemporary TAPs. In contrast, Yao *et al.* [36] presented an approach that introduced an interactive element to semantic analysis. They relied on a Hierarchical Reinforcement Learning framework to translate natural language descriptions into IFTTT rules. The approach involved training an agent with a hierarchical policy to maximize parsing accuracy while minimizing the number of questions asked to the user, making the process more efficient and user-friendly. Additionally, Huang *et al.* [18] conducted an in-depth analysis of the potential implications of incorporating natural language interfaces to assist users in customizing and automating their personal devices. They introduced *InstructableCrowd*, a crowd-powered system enabling users to program their devices via a natural language interface. The system focuses on creating simple programs that are easy to use and employs human crowd workers to operate the natural language interface. By incorporating more than one sensor/effector, InstructableCrowd addresses key problems with device customization and automation, offering a promising approach for programming devices in the future.

Unlike the approaches that concentrate on generating executable rules from natural language descriptions [12, 21, 27, 36, 37], or aim to enhance the rule definition process through user interactions [18], we address a different problem. We focus on checking the semantic consistency of a UDD against the actual rule behavior before its dissemination. This ensures that the UDD accurately represents the intended behavior of the rule and reduces the risk of misleading or deceptive rule descriptions.

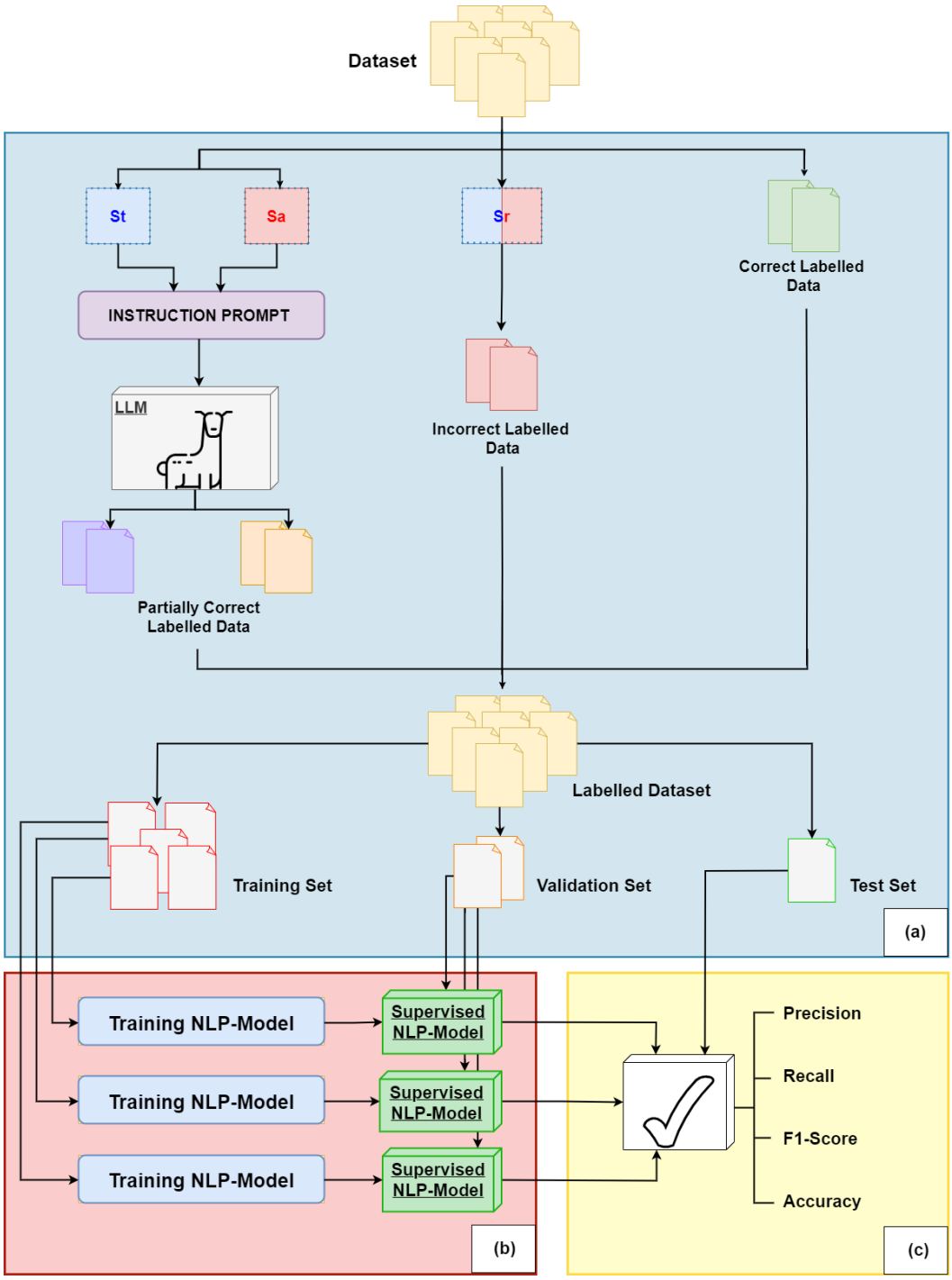


Figure 2: Proposed process for constructing the dataset and evaluating NLP-based models for checking the semantic consistency of a rule's UDD with respect to its trigger-action components. (a) Dataset construction. (b) Models training. (c) Models testing.

3. Methodology

In this section, we outline the approaches undertaken to build a comprehensive dataset encompassing UDD samples and to identify the most suitable NLP classification model for our specific objectives. In particular, Figure 2 illustrates the step-by-step process leading to the creation of the dataset and the establishment of effective supervised models to accurately examine the semantic relationships between UDDs and the synthesized patterns of their trigger-action components.

This process involves three main phases:

a) *Building and Labeling the Dataset*: The primary objective of this step is to prepare the labeled dataset required for training the NLP classification models. However, before initiating this procedure, it is essential to establish a pattern that effectively synthesizes

the behavioral aspects of a rule based on its trigger-action components. This pattern will serve as a reference for analyzing the corresponding UDD.

Furthermore, it is imperative to define the possible consistency classes for the UDD-pattern pairs and their associated labels. These classes categorize the UDDs into different types based on their alignment with the synthesized patterns. To achieve this, we first worked with the original dataset and partitioned it into four distinct samples. Each sample corresponds to different types of UDDs that a user might define for their rule, resulting in a diverse representation of descriptions across various consistency scenarios. By doing so, we ensure that each sample represents a consistent class, simplifying the labeling process significantly.

To account for various real-world scenarios, it was crucial to introduce shuffling mechanisms during the construction of three of these samples. These shuffling mechanisms (depicted as "*Sr*", "*St*", and "*Sa*" in Figure 2) introduce negative and partially negative samples of UDDs, considering that users might define descriptions that are inconsistent with the trigger component, the action component, or both. As a result, the shuffling mechanisms randomly modify one or both components of the rules in these samples, allowing us to incorporate samples of inconsistent and partially inconsistent UDDs.

Finally, all the samples are consolidated to create a labeled dataset that includes all types of samples required for training the NLP classification models. The dataset now comprises a large set of rules labeled according to their respective consistency classes, providing a solid foundation for the subsequent model training and evaluation stages.

b) *Training NLP Classification Models*: This stage focuses on training classification models using the labeled UDD-pattern pair dataset, referred to as the training set. The features used for training the models encompass the textual representation of the UDD, along with the corresponding synthesized pattern. By leveraging NLP techniques, we can extract crucial semantic information from these components, which the classification models can utilize to discern and distinguish among various consistency classes.

To ensure the effectiveness and accuracy of the classification models, we carried out a meticulous phase of dataset construction. As a result, we achieved a balanced training set, wherein each consistency class is evenly represented by 5,000 samples. This balanced distribution eradicates the issue of some classes being more frequent than others, preventing potential bias and ensuring that the models are equally well-trained on all consistency scenarios.

c) *Testing NLP Classification Models*: In this phase, our main objective is to thoroughly evaluate the performance of the NLP classification models. This evaluation is accomplished by providing the classification models with a carefully selected set of labeled UDD-pattern pairs, which serves as the input for testing their capabilities in determining semantic consistency.

To measure the effectiveness and accuracy of the models, we employ well-known evaluation metrics, i.e., Precision, Recall, F1-score, and Accuracy. By using these widely recognized evaluation metrics, we can effectively assess and compare the performance of the NLP classification models in the context of semantic consistency checking. These metrics provide valuable insights into the models' strengths and weaknesses, allowing us to make informed decisions about their suitability for real-world applications.

In the following sections, we provide a comprehensive description of the steps involved in constructing the labeled dataset tailored to serve our specific objectives. This dataset forms a crucial foundation for our research, facilitating the training and evaluation of the NLP classification models employed in our task.

4. Dataset Construction

As mentioned above, this section aims to outline the process of dataset construction and labeling, which serves as the foundation for training classification models dedicated to assessing the semantic consistency between the trigger-action components of an IFTTT rule and the accompanying natural language description. Our focus is on presenting both the starting dataset employed during analysis and the technical mechanisms involved in generating a new dataset specifically tailored for semantic consistency evaluation.

4.1. IFTTT Rule Dataset

In our study, we utilized the dataset proposed by *Mi et al.* [23], which provides a collection of IFTTT rules obtained from crawling the IFTTT.com website. The dataset contains important information such as the rule's title (*Title*), a description explaining the rule behavior (*Desc*), the event that triggers the rule (*TriggerTitle*) defined through a specific channel (*TriggerChannelTitle*), the action to be performed (*ActionTitle*) selected from the corresponding channel (*ActionChannelTitle*), and the name of the rule creator (*Creator Name*).

Exploiting the valuable information provided by IFTTT, we embarked on a novel approach for building a new dataset by devising a specialized pattern for *synthesizing UDDs*.

These patterns were meticulously designed to ensure a coherent and precise representation of a rule's behavior, capturing essential details about its trigger and action components presented in the original dataset. These structured patterns played a pivotal role in our research, serving two significant tasks.

The first task centered around the core objective of our study: the semantic consistency checking task. By employing the synthesized patterns, we were able to assess the se-

mantic alignment between a rule's trigger-action components and the corresponding UDD.

The second task involved harnessing the potential of Large Language Models (LLMs). By generating new random patterns using our structured approach, we presented these patterns as input to the LLM. This enabled us to generate samples of erroneous descriptions that users might write, encompassing incorrect triggers, actions, or even both components. This allowed for a comprehensive examination of potential user errors, broadening the scope of our analysis beyond just consistent samples.

In the final stage, we carried out *dataset labeling* by categorizing each UDD-pattern pair into its defined consistency class. This critical process significantly enhanced the efficiency of data organization and analysis. By accurately labeling the pairs, we ensured that our dataset encompassed a diverse representation of consistency scenarios, including complete consistency, complete inconsistency, and partial consistency of a UDD.

4.2. Synthesizing a UDD from the components of a rule

In our earlier study, we devised a structure that functions as a natural language description to evaluate the coherence between a UDD and the real behavior of a rule. This structure incorporates essential rule elements such as *trigger*, *trigger channel*, *action*, and *action channel*. The specific pattern used for generating the synthesized UDD is as follows:

> ***IF*** *TriggerTitle (TriggerChannelTitle)* ***THEN*** *ActionTitle (ActionChannelTitle)*

The adoption of this standardized format offers a concise and comprehensive representation of the core elements and occurrences associated with a specific rule. To illustrate this, we provide an example using an IFTTT rule that consists of the following components:

- ***TriggerTitle***: "Any new SMS received"
- ***TriggerChannelTitle***: "Android SMS"
- ***ActionTitle***: "Send me an email"
- ***ActionChannelTitle***: "Email"

The synthesized pattern for this rule is as follows:

> ***IF*** *Any new SMS received (Android SMS)* ***THEN*** *Send me an email (Email)*

This pattern offers a succinct and clear representation of the rule's components and their corresponding values, making it easy to understand the intended functionality. This holds true even after examining the original description:

> *When a text message arrives, forwards it to your email.*

4.3. Sample Generation

To configure an effective model, we undertook the creation of a new dataset, encompassing all possible types of UDDs that a user could generate for their automation rules. These UDD types fall into three distinct macro categories:

- ***Completely Consistent UDDs***: These UDDs are coherent, aligning perfectly with both the trigger and action components.
- ***Completely Inconsistent UDDs***: In contrast, these UDDs lack coherence with both the trigger and action components.
- ***Partially Consistent UDDs***: This category includes UDDs that exhibit coherence with either the trigger component or the action component, but not both.

To achieve this UDD diversification, we utilized the synthesizing strategy described in the previous section. Before delving into the strategy's adoption, we first defined the desired final structure of the dataset. Our objective was to construct a random sample of 20,000 entries, each comprising three key features: the UDD (description component), the synthesized pattern, and the corresponding label indicating the class of consistency between the UDD and its pattern.

After this, we embarked on generating different types of UDDs. For the first type of UDD (completely consistent), we randomly selected 5,000 rules with consistent descriptions from the initial dataset and synthesized their corresponding patterns from the rule's components. For the second type (completely inconsistent), we took 5,000 random rules and replaced each one's correct description with a different description that includes a distinct combination of trigger and action components. This shuffling mechanism introduces inconsistencies in the UDDs. It is important to note that the shuffling is applied only to the UDDs, while the patterns are synthesized with the correct components of the user's rule.

However, the most challenging aspect was defining samples for the last type of UDDs (partially consistent). To address this, we selected 10,000 random rules from the original dataset and divided them into two separate sets. This division enabled us to obtain 5,000 patterns with only the wrong trigger component (randomized from the other trigger components of the dataset) and another 5,000 patterns with only the wrong action component (randomized from the other action components of the dataset). These newly construed patterns were then used as input for the LLM Alpaca-Lora [2] [17, 32] to generate partially correct UDDs.

We adopted this approach due to the difficulty in designing precise instructions for a LLM. Instead, we opted for a single prompt, instructing the model to generate a textual description explaining the behavior of the automation rule

[2] https://crfm.stanford.edu/2023/03/13/alpaca.html

based on the input pattern containing the trigger and action components. The instruction prompt is the following:

> *Given an input sentence containing the trigger and action components of a trigger-action rule, execute the following instruction:*
> *"Generate a textual description explaining the behavior of the trigger-action rule."*

For example, given the following partially consistent pattern:

> ***IF** Current condition changes to (RSS Feed), **THEN** Post a tweet (Twitter)*

The model generates the corresponding UDD:

> *When the current condition changes to RSS Feed, a tweet is posted on Twitter.*

This prompt was used for both tasks, streamlining the process and ensuring consistent results. It is worth noting that, in the final dataset, we have the patterns synthesized with the correct components of the user's rule. The described process was solely for defining samples of partially correct user descriptions, and the generated UDDs were manually checked to ensure their correctness.

By incorporating these techniques and expanding the dataset to include a wide range of UDD types, we achieved a more robust and accurate set of data for the training phase.

4.4. Data Labeling

For the final stage of dataset composition, we proceeded to assign the appropriate labels to the UDD-pattern pairs. Building on our previous work [6], we had initially defined two semantic similarity classes:

- ***Contradiction***: This label denotes inconsistency between the UDD and the synthesized pattern, indicating that the UDD and the pattern do not align in their descriptions.
- ***Entailment***: This label signifies consistency between the UDD and the synthesized pattern, implying that the UDD's description is in agreement with the pattern.

However, this approach resulted in the exclusion of some possible scenarios, specifically the partial consistency UDDs, from the definition of a trigger-action rule description.

To address this limitation, we decided to define more appropriate labels for UDD-pattern pairs, taking into account the internal division of the dataset. As a result, we delineated the following four classes:

- ***"ee"***: This class denotes complete consistency between the UDD and the synthesized pattern, indicating that both the trigger and action components are accurately represented in the UDD.
- ***"cc"***: This class denotes complete inconsistency between the UDD and the synthesized pattern, indicating that neither the trigger nor the action components are correctly aligned in the UDD.
- ***"ec"***: This class denotes partial consistency between the UDD and the synthesized pattern, with a focus on the trigger component. Specifically, the trigger component in the UDD is correct, but the action component does not align with the pattern.
- ***"ce"***: This class denotes partial consistency between the UDD and the synthesized pattern, with a focus on the action component. Specifically, the action component in the UDD is correct, but the trigger component does not align with the pattern.

With the generation of our new dataset, we no longer require manual labeling since we have systematically produced the samples. Through the devised strategy and patterns, we are able to create diverse UDD-pattern pairs, covering various consistency scenarios, including completely consistent, completely inconsistent, and partially consistent cases.

The use of the LLM Alpaca-Lora enabled us to generate new UDDs that mimic user-generated descriptions with errors or partial consistencies. This procedure proved especially beneficial for creating UDDs falling into the "ec" and "ce" classes, where either the trigger or the action component was accurately represented, but the other exhibited inconsistencies. By automating the sample generation process, we have significantly reduced the manual effort involved in labeling the data. The resulting dataset contains a comprehensive representation of UDDs, covering a wide range of semantic similarities with their corresponding synthesized patterns.

With this augmented dataset, we can now proceed to train and evaluate our NLP classification models for the semantic consistency checking task more efficiently and effectively. The automated generation of samples not only saves time but also enhances the dataset's diversity, contributing to the overall robustness and accuracy of the trained models.

5. Classification Models

This section details the implementation of models for classifying the UDD-pattern pairs, the techniques used to develop each model, and the training phase setup.

We consider three different transformer-based models to assess the semantic consistency between UDDs and rule behaviors:

1. **BERT-based model**: This model is based on BERT [14]. The latter is an encoder-only model that utilizes deep bidirectional transformers and has been pretrained on a large corpus of data to create a powerful NLP language representation model.

2. **T5 model**: The T5 [29] model adopts a unique approach as an encoder-decoder model, learning to predict masked words in sentences through a corrupting span denoising objective.

3. **GPT-2 model**: The GPT-2 [28] model is an unsupervised generative language model developed by OpenAI. It operates as a decoder-only model based on the transformer architecture.

By employing these three transformer-based models, we aim to thoroughly analyze and compare their performance in determining the semantic alignment between UDDs and rule behaviors. The first model, based on the BERT architecture, involves an encoder-only approach and a feature extraction layer to represent input tokens. The second model, T5, utilizes both encoder and decoder components, offering insights into the performance of combined architectures. Finally, the third model, GPT-2, allows us to explore the capabilities of a purely generative approach.

In all models, features are treated as text. Moreover, before training the models, a pre-processing phase is performed to remove noise from UDDs. This includes operations such as normalization and lemmatization on the textual values.

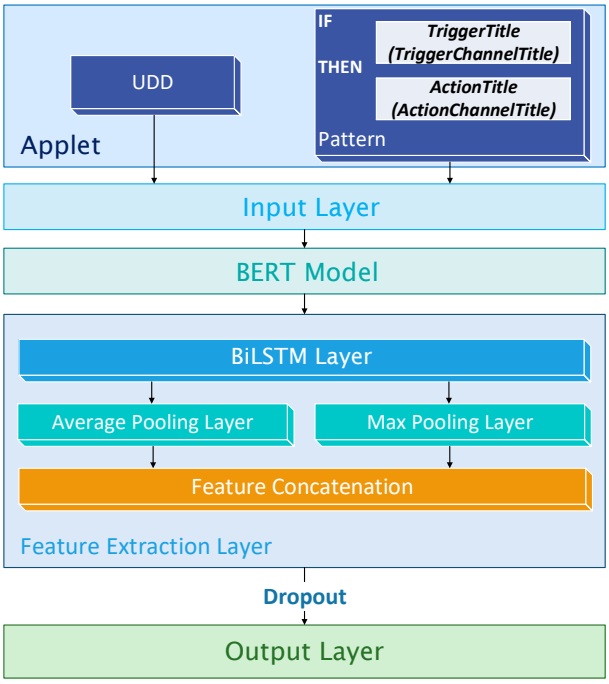


Figure 3: The architecture of the BERT-based model

5.1. BERT-based Model

The Bidirectional Encoder Representations from Transformers (BERT)-based model's architecture for classifying the semantic consistency of UDD-pattern pairs is illustrated in Figure 3. This model comprises interconnected components working together to achieve the objective. The initial component is the *Input Layer*, which encodes UDD-pattern pairs into numerical representations known as dense vectors. These dense vectors are then processed by the BERT language model, consisting of multiple Transformer Encoder Layers that generate contextual representations of each word in the input sequence using self-attention. Each layer produces dense vectors capturing different levels of syntactic and semantic information.

Next, the sequence obtained from the BERT model is passed to the *Feature Extraction Layer*, containing a *Bidirectional Long Short-Term Memory* (BiLSTM) Layer. The BiLSTM is designed to store past and future context, and its output consists of a sequence of vectors representing the hidden states at each time step. These hidden states are concatenated to create a representation capturing global features and dependencies.

The output from the BiLSTM Layer is processed through an *Average Pooling Layer* and a *Max Pooling Layer*, reducing dimensionality by aggregating information across the sequence. Average pooling calculates the average value of each feature, providing an overall representation and distribution. On the other hand, max pooling selects the maximum value from each dimension, highlighting salient features. Both pooling operations are concatenated through a *Feature Concatenation* module to create a comprehensive representation that captures overall context and important local details. A *Dropout* operation is applied before feeding the concatenated data to the *Output Layer*, randomly dropping out input features to mitigate overfitting.

The *Output Layer* utilizes the extracted features to evaluate semantic consistency between rule descriptions and corresponding patterns. It applies linear transformations to compute the final classification output of the model, determining whether the UDD-pattern pairs exhibit semantic consistency.

5.2. T5 Model

The architecture of the Text-To-Text Transfer Transformer (T5) model, designed specifically for the text classification task, is delineated in Figure 4. T5's approach to text classification is distinct from traditional methods that use separate encoder and decoder components. Instead, it transforms all tasks into a text-to-text format, where both the input and output are treated as text sequences. This allows T5 to handle different tasks by simply modifying the input and output representations.

The architecture begins with an *Input Layer* that takes the UDD-pattern pairs to be classified as input. The input text is processed and converted into a sequence of tokens, with each token representing a word or subword unit. These tokens are then embedded into a dense, continuous vector space using an *Embedding Layer*.

The embedded tokens enter the *Encoder Transformer Blocks*, which are responsible for processing the input text. Each encoder block consists of a multi-head self-attention layer, a feedforward neural network layer, and layer normalization. The self-attention layer allows the model to attend to different parts of the input text, capturing the relationships be-

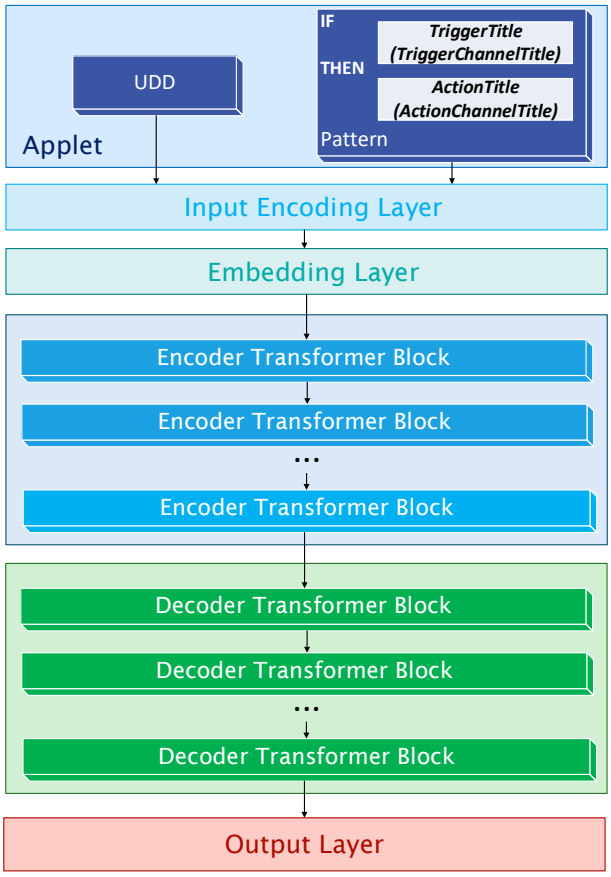


Figure 4: The architecture of the T5 model

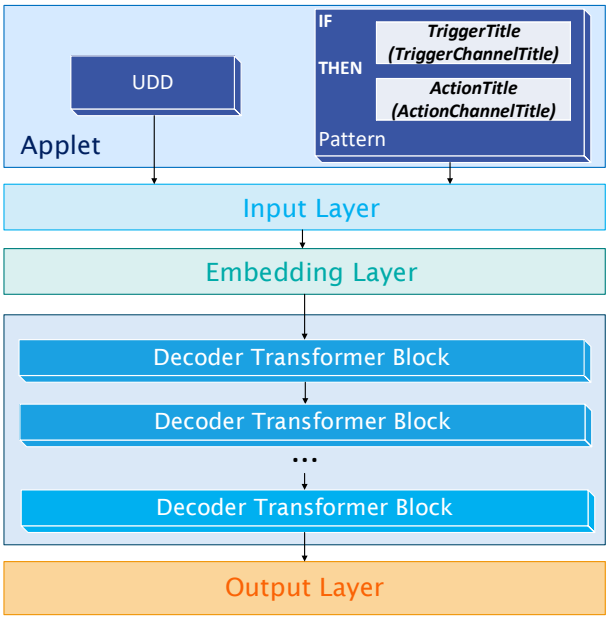


Figure 5: The architecture of the GPT-2 model

tween words and their context. The *Feedforward Neural Network* introduces non-linear transformations, and layer normalization helps stabilize the training process.

The output of the encoder blocks is a sequence of contextualized representations, with each token's representation containing information about its surrounding context. These representations are then passed to the *Decoder Transformer Blocks*. The latter further process the encoder's contextualized representations to generate task-specific outputs. Each decoder block has similar components to the encoder blocks, such as multi-head self-attention, Feedforward Neural Networks, and layer normalization. However, the decoder also includes cross-attention layers, allowing it to focus on both the input sequence and the task representation simultaneously.

Finally, the *Output Layer* receives the processed output from the last decoder block. This layer is customized to the specific text classification task and further processes the contextualized representations. It produces classification scores for each possible class, determining the predicted class for the input text based on the highest probability.

5.3. GPT-2 Model

The Generative Pre-trained Transformer 2 (GPT-2) model was originally designed for generating coherent and diverse text, but its capabilities have extended to include highly useful applications in text classification tasks as well.

The architecture of GPT-2, as depicted in Figure 5, is based on the transformer architecture, and it primarily leverages the decoder component of the transformer. Unlike encoder-decoder models such as T5, GPT-2 does not utilize the encoder part of the transformer. This design choice enables GPT-2 to excel in its primary function of generating text with contextual understanding.

In the context of text classification with UDD-pattern pairs as input, GPT-2's architecture begins with an *Input Layer*, which takes a sequence of tokens representing words or subword units. These tokens are then transformed into dense, fixed-dimensional vectors using an *Embedding Layer*. This mapping process places the tokens into a continuous vector space, facilitating the model's ability to capture semantic relationships and similarities between words.

The embedded tokens are then passed through a series of *Decoder Transformer Blocks*. Each transformer block is a stack of layers, consisting of a multi-head self-attention mechanism, which allows the model to attend to different parts of the input sequence and capture dependencies between words in the context of the entire sequence. Additionally, each block contains a *Feedforward Neural Network Layer*, which introduces non-linear transformations to the token representations, further enhancing the model's ability to model complex relationships.

In order to ensure stable training and facilitate faster convergence, each transformer block incorporates a normalization phase. This process normalizes the activations in each layer, enhancing the robustness of the optimization process.

As the input sequence progresses through the stack of transformer blocks, the model gains a deeper understanding of the context and relationships between the tokens. The fi-

nal transformer block produces a sequence of hidden states, where each hidden state corresponds to the encoded representation of its corresponding token.

For text classification tasks, the hidden states are then passed to the *Output Layer*, which performs a weighted combination of the hidden states and generates classification scores for each possible class. The class with the highest score is selected as the predicted class for the input text.

6. Models Evaluation

In this section, we present an analysis of the performances of the classification models. Specifically, we provide details on the experimental setup, the adopted metrics, and the results obtained from the experiments. The code of the software is publicly available on GitHub[3].

6.1. Evaluation Setup

In the evaluation phase, we trained the three NLP models using specific methodologies.

For the BERT-based model, we followed a two-step process. Initially, we froze all pre-trained layers and focused on training only the top layers. This allowed us to extract features by utilizing the representations of the pre-trained model. After feature extraction, we proceeded with an additional fine-tuning step. During this step, we unfroze the BERT model and retrained the entire architecture with a significantly low learning rate. The objective was to progressively adapt the pre-trained features to the new data, leading to enhanced model performance.

To pre-train and fine-tune the BERT-based model, we utilized Python libraries, specifically `Keras` and `TensorFlow`. We chose the "bert-base-uncased" variant, which has 12 transformer blocks, 768 hidden units, and 12 self-attention heads. It is designed to handle lowercase letters. The training set consisted of 13,999 samples, encompassing all four types of consistency classes. To optimize hyperparameters, we employed a validation set with 4,000 samples. The best hyperparameter configuration included 4 epochs, a batch size of 64, an epsilon set to 1e-5, and a maximum text length of 70. The model's performance was then evaluated using a test set of 2,001 UDD-pattern labeled pairs.

For GPT-2 and T5, we followed the same libraries and dataset sizes but made adjustments to hyperparameters. T5 was trained with 6 epochs, a batch size of 24, and a maximum text length of 70. On the other hand, GPT-2 was trained with 5 epochs, a batch size of 32, and a maximum text length of 60.

6.2. Evaluation Metrics

The performance evaluation of the proposed models involves several metrics, i.e., Accuracy, Precision, Recall, and F1-score. These metrics are computed based on the values of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The evaluation metrics can be expressed as follows:

[3] https://github.com/empathy-ws/TAP-Semantic-Consistency-Checking

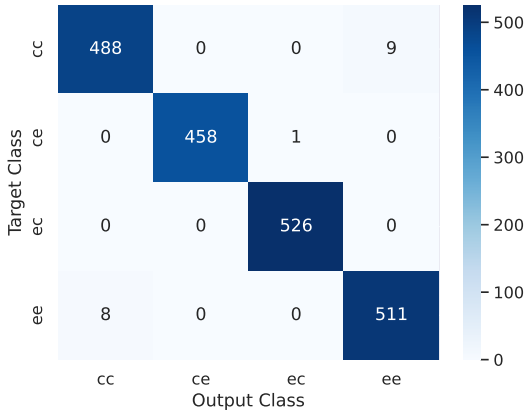


Figure 6: Confusion Matrix of the BERT-based model

- **Accuracy** is a measure of the overall correctness of a model's predictions, expressed as the ratio of the number of correctly classified samples to the total number of samples evaluated:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

- **Precision** is a measure of the proportion of true positive samples among all samples that the model identified as positive:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

- **Recall** is a measure of the proportion of true positive samples among all actual positive samples:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

- **F1-score** is the harmonic mean of Precision and Recall:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

6.3. Results and Discussion

In our investigation, we leveraged three transformer-based NLP models to evaluate the semantic consistency between UDDs and rule behaviors. The confusion matrices obtained from the classification results on the test set for each classification model are presented in Figure 6, Figure 7, and Figure 8, respectively. Additionally, Table 1 provides the resulting values of Accuracy, Precision, Recall, F1-score, and the average of the per-class metrics for each model.

The BERT-based model achieved the highest accuracy and precision values (99%), indicating its superior ability to make correct predictions overall. It excelled in identifying the ee and ec classes with high recall rates but encountered challenges in distinguishing class cc, leading to some misclassifications where ee was incorrectly predicted as cc. On

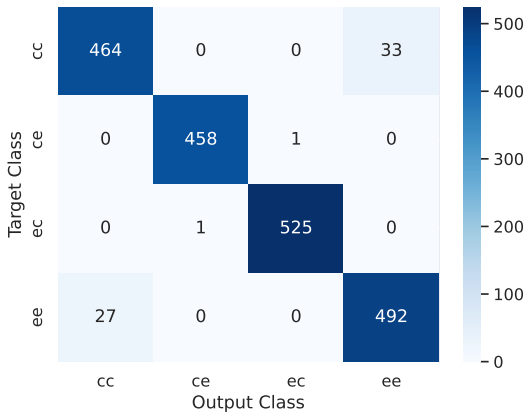


Figure 7: Confusion Matrix of the T5 model

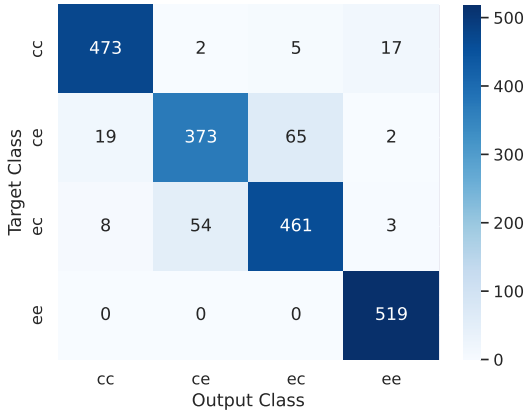


Figure 8: Confusion Matrix of the GPT-2 model

the other hand, the T5 model achieved an accuracy of 97%, exhibiting strong overall performance. Specifically, it performed well in identifying the ee and ec classes, evident from their recall values (Table 1). However, akin to the BERT-based model, the T5 model encountered difficulties concerning class cc, frequently leading to erroneous classification as an ee class. Finally, the GPT-2 model achieved an accuracy of 91% and exhibited the lowest discriminative ability among the three models. As highlighted by the confusion matrix (Figure 8), it faced notable challenges in differentiating class cc from other classes, increasing confusion between ec and ce. Despite these limitations, the GPT-2 model still produced mostly correct predictions.

The superior performance of the BERT-based model over the T5 and GPT-2 models can be attributed to the fundamental differences in their pre-training approach. In particular, BERT leverages the Masked Language Modeling (MLM) [14] technique during pre-training, where certain words in the input text are randomly masked, and the model is tasked with predicting these masked words based on contextual cues from the surrounding words. This process equips BERT

Table 1
Classification performances of the models on the test set

	Metric	cc	ce	ec	ee	Avg
BERT						
	Precision (%)	98	100	100	98	99
	Recall (%)	98	100	100	98	99
	F1-score (%)	98	100	100	98	99
	Accuracy (%)					99
T5						
	Precision (%)	95	100	100	94	97
	Recall (%)	93	100	100	95	97
	F1-score (%)	94	100	100	94	97
	Accuracy (%)					97
GPT-2						
	Precision (%)	95	87	87	96	91
	Recall (%)	95	81	88	100	91
	F1-score (%)	95	84	87	98	91
	Accuracy (%)					91

with a solid capability for acquiring contextual representations of words and comprehending intricate relationships between them within sentences. On the contrary, T5 adopts an innovative text-to-text approach [29] during its pre-training phase, where the input text serves as a description of a specific NLP task, while the output text represents the corresponding solution or result for that particular task. When focusing specifically on classification tasks, the MLM approach of BERT exhibits notable advantages. By predicting masked words, BERT gains a deeper understanding of how words are interconnected within a given context, resulting in the proficient classification of text. Instead, formulating classification tasks into the text-to-text format for T5 might not be as straightforward as employing BERT's masked language modeling. Indeed, ensuring that the task descriptions lead to accurate and effective classification can pose challenges and may necessitate meticulous formulation and experimentation to achieve optimal results. Furthermore, it is essential to consider the trade-off between specificity and generalization. While BERT's MLM approach enables it to focus on the contextual nuances of individual words, T5's text-to-text approach emphasizes generalization across diverse tasks. As a consequence, T5 might not capture certain task-specific nuances as effectively as BERT in certain classification scenarios. Finally, unlike the BERT and T5 models, the GPT-2 model adopts a unidirectional approach (i.e., it processes tokens in a left-to-right manner), which can result in a less comprehensive understanding of the input text. This unidirectional nature may have impacted its ability to effectively differentiate between the classes, particularly in cases where the context from the right side of the input text was crucial for accurate prediction.

The promising performance of models in checking semantic consistency between UDDs and rule behavior presents opportunities for further research and improvements in transformer-based NLP classification for similar tasks. Understanding the strengths and weaknesses of different transformer

models helps inform the selection of appropriate models based on the task requirements. As transformer-based NLP models continue to advance, they hold great potential in enhancing the accuracy and efficiency of various natural language understanding tasks, benefiting a wide range of applications in the domain of IoT device automation and beyond.

7. Conclusion and Future Work

This paper introduces an innovative approach to tackle the issue of potential inaccuracies and misleading information in UDDs shared on TAPs. We developed a new dataset that comprehensively covers different types of UDD scenarios, including consistent, inconsistent, and partially consistent descriptions. To achieve this, we leveraged an LLM to generate samples with partially correct UDDs, which reduced the manual workload and increased dataset heterogeneity.

The evaluation involved three NLP classification models: BERT-based, T5, and GPT-2. The BERT-based model underwent a two-step training process, where initially, the top layers of the pre-trained model were targeted for training, and then fine-tuning was conducted with the entire architecture. The T5 and GPT-2 models were also trained using the same dataset size as BERT. The experimental results on the IFTTT dataset, consisting of 20,000 labeled UDD-pattern pairs, demonstrated the effectiveness of the proposed models. The BERT-based model demonstrated remarkable performance, achieving an overall accuracy of 99%, complemented by precision, recall, and F1-score, all attaining a value of 99% as well. In contrast, the T5 model exhibited slightly inferior performance, with all evaluation metrics registering at 97%. The GPT-2 model yielded the least favorable results, scoring 91% for all evaluation metrics. This outcome accentuates the relatively weaker performance of the decoder-only architecture-based model in performing the classification task. Nevertheless, it is noteworthy that all models exhibited a high degree of reliability in discerning compliant UDDs from unrelated ones, establishing a robust and dependable method for conducting semantic consistency checks in TAPs.

In the future, we would like to consider the adoption of this approach in other TAPs beyond the IFTTT case study. This broader application would enable us to further validate the robustness and effectiveness of our classification models in different environments and contexts. Additionally, we aim to explore potential enhancements to the models, such as incorporating more advanced NLP techniques or leveraging larger and more diverse datasets for training. Furthermore, we believe that integrating user feedback and iterative improvements to the models would be valuable in optimizing the accuracy and reliability of the semantic consistency checking process. In addition, providing explainability to users has already been proven to increase users' trust in the systems in several domains [4, 8]. Thus, we think incorporating such a module would allow us to contribute to the evolution of TAPs, in terms of trust, security, and ease of use in defining automation rules while safeguarding users against potential risks arising from misleading or deceptive rule descriptions.

References

[1] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., Zhou, Y., 2017. Understanding the Mirai Botnet, in: Proceedings of the 26th USENIX Conference on Security Symposium, USENIX Association, USA. p. 1093–1110.

[2] Breve, B., Cimino, G., Desolda, G., Deufemia, V., Elefante, A., 2023a. On the user perception of security risks of tap rules: A user study, in: International Symposium on End User Development, Springer. pp. 162–179.

[3] Breve, B., Cimino, G., Deufemia, V., 2021. Towards a classification model for identifying risky IFTTT applets, in: Proceedings of the 2rd International Workshop on Empowering End-Users in Dealing with Internet of Things Ecosystems, pp. 33–37.

[4] Breve, B., Cimino, G., Deufemia, V., 2022. Towards explainable security for ECA rules, in: Proceedings of the 3rd International Workshop on Empowering End-Users in Dealing with Internet of Things Ecosystems.

[5] Breve, B., Cimino, G., Deufemia, V., 2023b. Identifying security and privacy violation rules in trigger-action IoT platforms with NLP models. IEEE Internet of Things Journal 10, 5607–5622.

[6] Breve, B., Cimino, G., Deufemia, V., Elefante, A., 2023c. A BERT-based model for semantic consistency checking of automation rules. Proceedings of the 29th International DMS Conference on Visualization and Visual Languages .

[7] Breve, B., Cimino, G., Deufemia, V., Elefante, A., 2023d. On privacy disclosure from user-generated content of automation rules, in: Joint Proceedings of the Workshops, Work in Progress Demos and Doctoral Consortium at the IS-EUD 2023, pp. 1–5.

[8] Cascone, L., Pero, C., Proença, H., 2023. Visual and textual explainability for a biometric verification system based on piecewise facial attribute analysis. Image and Vision Computing 132, 104645.

[9] Celik, Z.B., Tan, G., McDaniel, P.D., 2019. IoTGuard: Dynamic enforcement of security and safety policy in commodity IoT, in: Network and Distributed System Security (NDSS) Symposium.

[10] Chiang, Y.H., Hsiao, H.C., Yu, C.M., Kim, T.H.J., 2020. On the privacy risks of compromised trigger-action platforms, in: Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part II 25, Springer. pp. 251–271.

[11] Cobb, C., Surbatovich, M., Kawakami, A., Sharif, M., Bauer, L., Das, A., Jia, L., 2020. How risky are real users' IFTTT applets?, in: Proceedings of the Sixteenth USENIX Conference on Usable Privacy and Security, pp. 505–529.

[12] Corno, F., De Russis, L., Monge Roffarello, A., 2020. HeyTAP: Bridging the gaps between users' needs and technology in IF-THEN rules via conversation, in: Proceedings of the International Conference on Advanced Visual Interfaces, pp. 1–9.

[13] Desolda, G., Ardito, C., Matera, M., 2017. Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools. ACM Transactions on Computer-Human Interaction (TOCHI) 24, 1–52.

[14] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .

[15] Ding, W., Hu, H., 2018. On the safety of iot device physical interaction control, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 832–846.

[16] Hsu, K.H., Chiang, Y.H., Hsiao, H.C., 2019. Safechain: Securing trigger-action programming from attack chains. IEEE Transactions on Information Forensics and Security 14, 2607–2622.

[17] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang,

L., Chen, W., 2021. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 .

[18] Huang, T.H., Azaria, A., Romero, O.J., Bigham, J.P., 2019. Instructablecrowd: Creating if-then rules for smartphones via conversations with the crowd. arXiv preprint arXiv:1909.05725 .

[19] Johnsson, B.A., Magnusson, B., 2020. Towards end-user development of graphical user interfaces for internet of things. Future Gener. Comput. Syst. 107, 670–680.

[20] Li, S., Xu, L.D., Zhao, S., 2015. The internet of things: a survey. Information systems frontiers 17, 243–259.

[21] Liu, C., Chen, X., Shin, E.C., Chen, M., Song, D., 2016. Latent attention for if-then program synthesis. Advances in Neural Information Processing Systems 29.

[22] Manaris, B., 1998. Natural language processing: A human-computer interaction perspective, in: Advances in Computers. Elsevier. volume 47, pp. 1–66.

[23] Mi, X., Qian, F., Zhang, Y., Wang, X., 2017. An empirical characterization of IFTTT: ecosystem, usage, and performance, in: Proceedings of the 2017 Internet Measurement Conference, pp. 398–404.

[24] Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N., Mian, A., 2023. A comprehensive overview of large language models. arXiv preprint arXiv:2307.06435 .

[25] Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., Ghani, N., 2019. Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations. IEEE Communications Surveys & Tutorials 21, 2702–2733.

[26] Paternò, F., Santoro, C., 2019. End-user development for personalizing applications, things, and robots. International Journal of Human-Computer Studies 131, 120–130.

[27] Quirk, C., Mooney, R., Galley, M., 2015. Language to code: Learning semantic parsers for if-this-then-that recipes, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 878–888.

[28] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al., 2019. Language models are unsupervised multitask learners. OpenAI blog 1, 9.

[29] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research 21, 5485–5551.

[30] Surbatovich, M., Aljuraidan, J., Bauer, L., Das, A., Jia, L., 2017. Some recipes can do more than spoil your appetite: Analyzing the security and privacy risks of IFTTT recipes, in: Proceedings of the 26th International Conference on World Wide Web, p. 1501–1510.

[31] Tian, Y., Zhang, N., Lin, Y.H., Wang, X., Ur, B., Guo, X., Tague, P., 2017. Smartauth: User-centered authorization for the internet of things, in: Proceedings of the 26th USENIX Conference on Security Symposium, USENIX Association, USA. p. 361–378.

[32] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al., 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 .

[33] Wang, Q., Datta, P., Yang, W., Liu, S., Bates, A., Gunter, C.A., 2019. Charting the attack surface of trigger-action IoT platforms, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA. p. 1439–1453.

[34] Wang, X., Li, J., 2013. Detecting communities by the core-vertex and intimate degree in complex networks. Physica A. 392, 2555–2563.

[35] Xiao, D., Wang, Q., Cai, M., Zhu, Z., Zhao, W., 2019. A3ID: an automatic and interpretable implicit interference detection method for smart home via knowledge graph. IEEE Internet of Things Journal 7, 2197–2211.

[36] Yao, Z., Li, X., Gao, J., Sadler, B., Sun, H., 2019. Interactive semantic parsing for if-then recipes via hierarchical reinforcement learning, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI Press.

[37] Yusuf, I.N.B., Jiang, L., Lo, D., 2022. Accurate generation of trigger-action programs with domain-adapted sequence-to-sequence learning, in: Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension, Association for Computing Machinery, New York, NY, USA. p. 99–110.

[38] Zeng, E., Mare, S., Roesner, F., 2017. End user security and privacy concerns with smart homes, in: Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017), USENIX Association, Santa Clara, CA. pp. 65–80. URL: https://www.usenix.org/conference/soups2017/technical-sessions/presentation/zeng.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Designing an Efficient Document Management System (DMS) using Ontology and SHACL Shapes[⋆,⋆⋆]

Maria Assunta Cappelli[a,*], Ashley Caselli[a,*] and Giovanna Di Marzo Serugendo[a,*]

[a]Centre Universitaire d'Informatique Université de Genève–CUI, Geneva, Switzerland

ARTICLE INFO

Article History:
Submitted 5.10.2023
Revised 7.31.2023
Accepted 8.2.2023

Keywords:
Automatic Document Processing
Data Management Systems
Knowledge Graph-based Approach
Reasoning Engine

ABSTRACT

Document management systems (DMS) are widely used for the management of business documents because they use metadata to organise and categorise digital documents. However, they are often based on unstructured and monolithic files and this structure raises questions about the quality and completeness of the information. To overcome this problem, this paper proposes a semantic rule-based approach for an RDF-based DMS, which uses a combination of ontology and Shapes Constraint Language (SHACL) rules to integrate legal aspects, validate data (expressed in an RDF triple store), reason and infer new information. The process is dynamic because the proposed DMS can automatically reason and create new inferences based on the information and data extracted from documents. The process also reasons on user profiles and underlying rules, capturing specific legal regulations, enabling further accurate and automated document management. The ontology used in the process captures specific concepts of Swiss tax returns, while the SHACL rules serve to reason about actual RDF triples relating to different tax households. The proposed DMS is innovative for its ability to reason on a specific domain, improve the accuracy and completeness of information managed. This work is relevant for any domain involving administrative documents and regulations (e.g. fiduciary or insurance sector).

1. Introduction

Organisations are using Document Management System (DMS) tools to simplify the management of their digital documents and files. A DMS is used to create, store, organise, retrieve, and update documents and files in a secure and efficient manner [8]. By using a DMS, organisations can streamline their document-dependent workflows, reduce manual document handling, and improve the accuracy and accessibility of their information.

A DMS can use metadata to organise and categorise documents. By tagging documents with metadata, users can easily search and retrieve documents based on specific criteria, such as date range, author, or content.

In the last ten years, the emergence of metadata-driven DMS platforms has transformed the way organisations handle their documents. These platforms have simplified the task of classifying, searching, and retrieving documents, resulting in enhanced productivity and collaboration among employees. Moreover, the adoption of cloud-based DMS solutions has enabled staff to access and work on documents from anywhere, at any time, further augmenting the efficiency and effectiveness of document-based workflows.

Efficient document management is crucial not only for the optimal retrieval and use of documents but also for effective and efficient work organisation. Indeed, Gorelashvili [6] notes that in the legal sector, automated document management is essential to improve and streamline the way lawyers manage their practice. DMS ensures that documents are easily accessible, well-organised, and protected. Abbasova [1] highlights the beneficial effects of DMS on workflow forms by automating the routing of documents between people, eliminating bottlenecks and optimising business processes.

⋆This document is the results of the research project n. 50606.1 IP-ICT "Addmin" funded by Innosuisse.

⋆⋆The present paper is an extended and revised version of the paper [3] presented at DMSVIVA 2023.

*Corresponding author

maria.cappelli@unige.ch (M.A. Cappelli); ashley.caselli@unige.ch (A. Caselli); giovanna.dimarzo@unige.ch (G.D.M. Serugendo)

ORCID(s): 0000-0001-8492-0354 (A. Caselli)

DOI reference number: 10.18293/JVLC2023-N2-034

DMS ensures more accurate organisation of business processes within the company through effective management and support of the quality system in accordance with inter-national standards, as well as efficient storage, management, and access to information and knowledge. Gostojić et al. [7] list five main benefits of DMS for organisations, including paper cost savings, more efficient use of space, and increased productivity. It also provides document security, easy access to documents, and a version control feature that allows ac-cess to previous versions. In addition, it provides damage control through backup creation and repository export and ensures consistency of procedures through protocol enforce-ment. Yousufi [19] discusses the benefits of a "paperless" workplace, where the use of paper is reduced through the digitisation of documents and the use of DMS. Among the benefits highlighted are the ability to save space, time and money, as well as improving document security and simpli-fying data transfer, and a positive impact on the environment. The paper production is associated with deforestation, the use of large amounts of water, and the production of green-house gas emissions.

Despite the benefits of using a DMS, some DMS solutions require significant setup processes and associated fees, or may not be well-suited to industry specific professions, creating further challenges for organisations seeking to implement an efficient DMS. In addition, since most solutions are enterprise-based, there are currently no solutions that automate access to critical documents for individual consumers. There is still no DMS that can classify, understand, and reason with customer documents, automatically process a bundle of customer documents and create a customer profile in compliance with regulations. In particular, in the domain of fiduciary, insurance brokerage, or tax returns, documents are still processed manually, sent by email or via a customer's cloud platform.

To overcome this problem this paper proposes a semantic rule-based approach for an RDF-based DMS. The proposed DMS uses a combination of various techniques: (1) an ontology for defining the concepts of the domain and their relationships (e.g. tax return); (2) data extracted from documents organised as RDF triples stored in a triplestore, following the ontology structure; and (3) Shapes Constraint Language (SHACL) rules for data validation, capturing regulations related to the domain, reasoning and inferring new information. The ontology organises the data in a structured way and defines the relationships between entities. This makes it easier to search and access documents. SHACL is an RDF-based rule specification language used to define shape properties, constraints and rules for data validation and verification [9]. Their combination allows the creation of a highly efficient, and automated DMS. In particular, the use of an ontology simplifies data organisation and management, while SHACL ensures data quality and reliability.

Besides providing a complete approach and a workflow, we addressed the specific case study of Swiss tax returns. We designed and implemented a rule-based process that dynamically builds, updates and reasons on users' profiles, integrating underlying legal regulations providing a DMS compliant by-design. An additional module completes this process by automatically extracting information from documents provided by users. Based on an ontology capturing the concepts of Swiss tax returns, we designed SHACL shapes to reason about asserted RDF triples of different tax households. A detailed description of the approach can be found in the technical report [4].

The remaining sections of the paper are organised as follows. Section 2 provides an overview of existing approaches in the context of DMS. Section 3 describes the proposed semantic-based approach in detail, our research questions, case study and workflow. The ontology used in the approach is discussed in section 4, and section 5 presents the rule-based methodology used to model the data. The implementation of the SHACL-based rules and their execution is shown in section 6, which includes details of how the rules were integrated into the system. Section 7 provides examples of validating RDF data against the defined SHACL shapes, and evaluating the defined rules. Finally, section 8 concludes the paper by summarising the main contributions of the proposed approach and discussing limitations and possible future research directions in the field of DMS.

2. Related work

Our proposed DMS system is innovative compared to other systems on the market because it uses a combination of semantic techniques (ontology and SHACL shapes), which allows even more precise and automated document management. In addition, the use of inference techniques allows new information to be derived from existing data, improving the accuracy and completeness of the information managed by the system. Therefore, we focus here on research works that either use ontologies or consider semantic approaches.

2.1. Ontology Approaches for DMS

Some researchers propose the use of ontologies as part of semantic document management approaches. Ontologies can formally define the structure, content and relationships of different types of documents. An ontology-based DMS can monitor document processes and workflows, track dependencies between documents, analyse how changes to one document may affect other related documents, and design the synchronisation steps needed to maintain consistency across interrelated document collections.

Fuertes et al. [5] develop an ontology for DMS concerning the construction sector. The ontology aims to classify documents along the lifecycle of the construction project, to reduce interoperability and information exchange problems, to establish a hierarchical structure of the different domains that correspond to the lifecycle of such projects, and finally to enable an interconnected system between these domains.

Doc2KG [16] is a framework that provides a continuous transformation of open data into a knowledge graph, using existing domain ontology standards. The system handles the initial conversion of a DMS into a knowledge graph and supports the continuous populating of the created knowledge

graph with new documents. The authors rely on a combination of natural language processing techniques to facilitate information extraction and constraint-solving techniques for knowledge graph creation and manipulation.

Lee et al. [10] develop a domain-specific ontology to support automatic document categorisation. The ontology contains a complete and detailed hierarchy of concepts used to represent documents related to information systems and technology as a set of concepts with relative weights. Although researchers recognise the advantages of using an ontology with classes in terms of the interpretability and comprehensibility of classification decisions, no reference is made to the definition of semantic rules to make the use of the ontology more flexible.

Sheng and Lingling [14] propose the use of ontology in the context of e-governance to model government data and create a semantic environment for managing government information. They present a semantic-based e-government system structure and use OWL as the ontology description language to provide the basis for data sharing and analysis. The authors detail the conceptual entity, the conceptual property and the relationship between concepts, which correspond to classes, properties and axioms respectively in the OWL language. However, they do not model semantic rules to define constraints and restrictions on the data, to ensure that it is consistent with the ontological structure they have defined, or to exploit its reasoning power.

Sladić et al. [15] present an approach to improve DMS through the use of a formal and explicit document model based on ontologies. This document allows the formal and explicit representation of the information contained in documents and the clear definition of concepts and relationships between them. In this way, the semantics of document content can be understood by machines, enabling more efficient and accurate analysis, classification and retrieval of documents. As a result, DMS can automatically classify documents based on their content, identify relationships between concepts, and support semantic search of documents.

2.2. Semantic Approaches for DMS

Wang et al. [18] organise and manage large amounts of documents through a representation of document semantics. The representation of document semantics is based on a set of attributes and a content vector, which allows for more accurate document identification and provides associative search capabilities. In addition, this study presents keyword-based indexing techniques and structural querying techniques for XML data, which are widely used for representing and exchanging data on the Web.

Amato et al. [2] propose a semantic analysis-based approach to make up for the lack of an adequate data structure of DMS. This lack can raise a problem for the application of appropriate security policies in DMS. The semantic methodology is able to retrieve information from specific parts of the document that can be useful for classification, security, etc. Semantic analysis serves for implementing fine-grained access control on sensitive data contained in unstructured and monolithic files, such as those found in DMS. The case study concerns the formalisation and protection of electronic health records.

Leukel et al. [11] propose a software architecture for cooperative semantic document management. They argue that semantic approaches to document management rely on enriching metadata and deriving semantic document models, but the quality of the metadata and the underlying domain ontology can limit the discovery of relationships between documents. The proposed software architecture aims to solve this problem by separating the semantic representation of individual documents from the knowledge of domain-specific relationships in two architectural layers.

Some of the above studies use ontologies to improve the effectiveness of DMS, while others use other semantic techniques. The former differ in their specific application domains and objectives, but they share the use of ontologies as a tool for semantic document management. These studies focus on semantic approaches to DMS but do not necessarily use semantic rules explicitly. These works often use techniques such as keyword-based indexing and structural retrieval, semantic analysis for sensitive data protection, or software architecture to improve the quality of metadata and domain ontologies. However, none of this research appears to use the combined approach of ontologies, and SHACL shapes, which can provide a more comprehensive and sophisticated DMS. Such an approach can ensure data quality and consistency while improving the effectiveness of document analysis, classification and retrieval.

3. Semantic-based approach

We discuss here our approach, starting with research questions, defining our case study, and providing our global workflow and architecture.

3.1. Research questions

The research questions proposed in this study relate to the processing of business documents, in particular administrative documents, for the company's customers. The aim is to propose a semantic rule-based approach that can help companies manage their customers' documents more efficiently and effectively. Our proposal addresses the following research questions:

A) How can a document be classified and multi-labelled based on extracted information that provides its key features?
B) How can customer profiles be created and updated based on the documents provided and the information extracted from them?
C) How can a reasoning process determine which documents customers need to provide based on their profiles?

3.2. Case study

In this paper, we focus on the Swiss tax return of households and the documents required to complete the tax return. We limit our case study to household profiles consisting of a

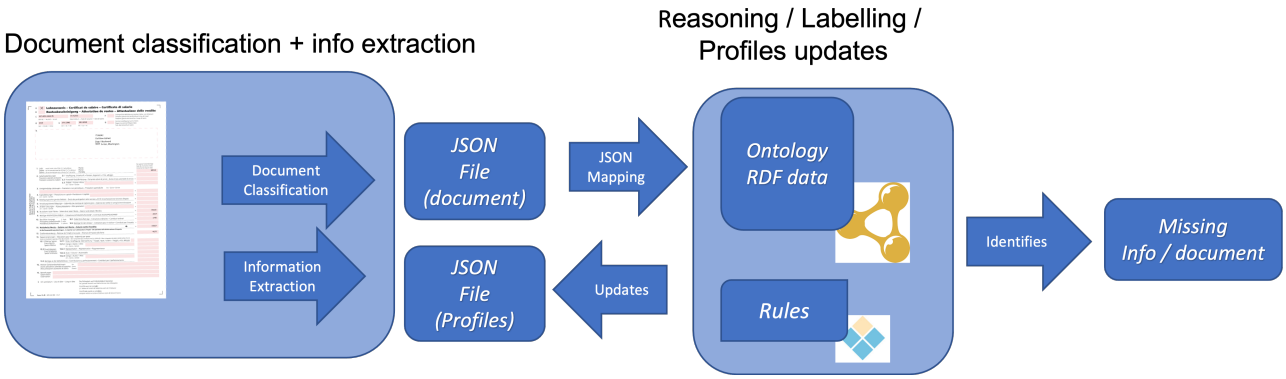


Figure 1: Overview of the Global Workflow as presented in [4]

single person, a widow/widower, couples, households with or without children or other dependants, retired or working. We have also limited our case study to the minimum set of administrative documents required to complete the Swiss tax returns of the households described above, namely: annual income, bank statements, health insurance policies and benefits, and family allowances, for each household member. We have included the health insurance statements because they are legally mandatory in Switzerland and everyone has to provide them for tax purposes.

Let us consider a tax household consisting of two working parents with children. As each parent is employed, data are extracted from their two salary certificates. The data extraction process identifies the main features of the document that are necessary conditions for a document to be classified as a wage (revenue) statement.

In response to the first research question A), shape properties and rules are applied to classify documents as salary certificates based on the extracted features. The system then assigns a double label (or tag) of "Tax" and "Income" to the salary statement.

In response to the second research question B), the system profiles both parents as employees.

Finally, in relation to the third research question C), the system identifies other necessary documents that the two parents and their children need to provide, such as health insurance.

3.3. Global Workflow and Architecture

Figure 1 shows the global overview of the workflow of our approach, which is described in detail in [4].

Such an architecture consists of three modules:

1) The documents (native PDFs or scanned documents) are processed by a *Document Classification and Information Extraction Module*. This module generates *JSON files for each document*, identifying its classes (e.g. health insurance policy, etc.), as well as specific information extracted from the document, such as date and amount.
2) The information extracted from the documents is also used to feed *JSON files profiles* of the household and its various members (e.g. widow/er, child, etc.);
3) The JSON information is then mapped to RDF by the *Reasoning, Labelling and Profiles Updates* module, using an ontology for Swiss tax returns and personal profiles. This module also contains a semantic rule-based reasoner, which is used on the one hand to update the information in the profiles (e.g. health insurance for a new child means that the child must be added to the household, possibly changing the household profile from a couple without children to a couple with children), and on the other hand to identify any missing documents of the household based on the existing profiles (e.g. health insurance or benefits are missing for a person identified as part of the household).
The *Reasoning, Labelling, and Profiles update* module has the following components: *(i)* an *ontology for managing Swiss tax and administrative documents*, based on actual official legal tax documents and on actual administrative documents needed to complete the tax declaration. The ontology defines concepts such as documents, user profiles, tax items, and changes in residence and status; *(ii)* The *rules* defined for validating documents, updating profiles based on new information, labelling documents, identifying missing documents (e.g. not provided in the bundle), integrating legal regulations aspects; and *(iii)* the *RDF data* mapped from the JSON files (actual data) containing information automatically extracted from the documents using an information extraction module.

4. Ontology for the management of tax and administrative documents

The ontology for the management of tax and administrative documents is the foundation for the DMS. It includes classes such as tax and administrative documents, user profiles, tax items and changes of status and domicile. The ontology has been developed in French[1] through Protégé [12].

The ontology creation process was based on a middle-out approach [17] because of the limited number of documents

[1]The ontology is available (on request) at the following website https://gitlab.unige.ch/addmin/doc-onto/-/wikis/home

available at the beginning of the project. The middle-out approach allows us to start with a limited set of data and then gradually expand the system as we acquired more information. This approach also ensures to adapt the system to user needs and to improve its accuracy over time. The step-by-step development process involved analysing tax return forms from 2020, as well as the instructions for filling them out issued by the Canton of Geneva, and model forms from other documents such as salary statements, 2nd and 3rd pillar pension funds, health insurance, etc. The results of the step-by-step development process were further validated by tax experts to ensure their validity and consistency over time.

The middle-out approach was applied as follows. First, we identified the basic concepts of the domain in terms of *(i)* documents and *(ii)* user profiles. We then developed two further parts for specific sub-areas, such as *(iii)* a section for tax items, representing the different categories of taxes and fees that the taxpayer has to pay, and *(iv)* a section for managing changes of status and domicile (address) of the user. These parts were then integrated into a larger and more complex ontology for the domain of tax and administrative documents.

To facilitate the integration of the different parts, we used SHACL shapes to define the relationships between the RDF nodes to ensure that the parts were correctly and consistently integrated into the larger ontology (section 6).

Figure 2 shows the middle-out process for generating the ontology.

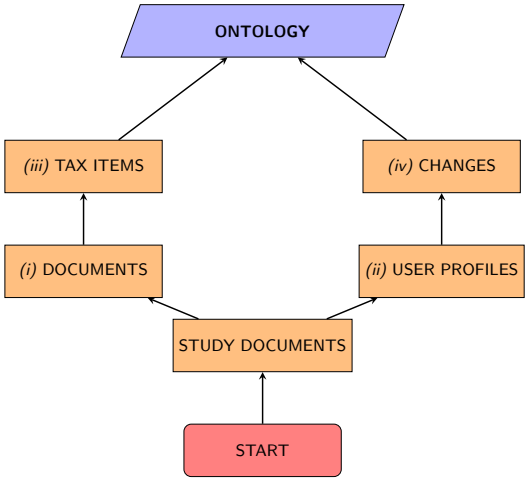


Figure 2: The development of the ontology

The middle-out approach takes into account real-world details, allowing specific domain information to be incorporated into ontological models. Finally, the approach helps to reduce the risk of instability and inconsistencies and ensures that the system can be tested and user feedback can be gathered more quickly as it is developed incrementally.

The results of the ontology development process are summarised in figure 3, which shows an extract of the ontology metrics, including 240 classes, 24 data type properties, 613 axioms, and 15 object properties.

Ontology metrics:

Metrics	
Axiom	613
Logical axiom count	327
Declaration axioms count	279
Class count	240
Object property count	15
Data property count	24

Figure 3: Ontology metrics

4.1. Ontology - Swiss Tax return

The ontology was developed following the steps (proposed by Noy and McGuinness [13]) to describe the domain of document management for tax and administrative returns. The following steps are followed:

1) **Defining the domain and scope of the ontology**.
In order to define the scope of the ontology and its functionalities, we asked questions such as: What tax and administrative documents are relevant to a particular tax return? How should these documents be organised and archived? What metadata is associated with the documents? What are the relationships between the documents, such as hierarchical dependencies between documents? What data needs to be extracted from the documents? Which household and user profiles are relevant for tax return? Answering these questions helped us to define the scope of the ontology and to identify its main functionalities.

2) **Defining the classes and the class hierarchy**.
The top-level classes of the ontology are: *(i)* documents, *(ii)* user profiles, *(iii)* tax items, *(iv)* changes. They have several middle-level classes. These middle-level classes are then further organised into sub-classes. Defining middle-level classes helps to make the ontology more understandable. It also allows to search and retrieve information from the ontology, as users can navigate through the hierarchy to find the information they need.

   (i) Documents. Within our ontology, we have defined a hierarchy of classes for fiscal and administrative documents. The top-level class Document represents the parent class of all other classes, including: “Copie”, “Attestation”, “Bordereau”, “Certificat”, etc. Additionally, we have defined further sub-classes for each of these classes to organise the different types of documents in a hierarchical manner, as shown for the class “Certificat” in figure 4. Among these, we distinguish for instance the Salary Certificate “CSalaire”, which is the main document for income report.

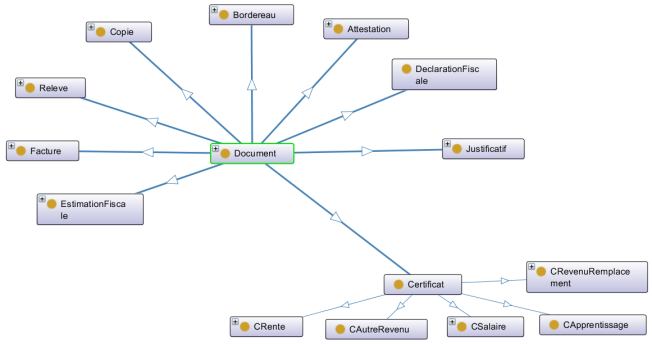


Figure 4: The section of *Documents* within the Ontology (figure adapted from Protégé ontology editor tool

(ii) User profiles. We identified the user profiles on the basis of a set of relevant and significant criteria for the ontology of the tax and administrative documents concerned. We chose the following criteria: type of employment, ownership of real estate, financial accounts, or movable property (e.g. car), any dependant person linked to the household, and civil status as shown figure 5.

The criteria used to identify user profiles were chosen on the basis of their relevance and importance for the ontology of the tax and administrative documents in question. The employment type criterion was used because the applicable tax regime and the required documents may vary depending on the employment status of the user. For example, an employee may have different tax documents than a self-employed person or an entrepreneur. We selected the criterion of Ownership of real estate or financial accounts, because ownership of such property may affect the user's tax situation. For example, the ownership of real estate may require the submission of specific documents for tax declaration. Similarly, the criterion of the number of dependants affects the user's tax situation and the documents required. For example, the presence of dependant children requires the submission of specific documents in order to obtain tax benefits. Finally, we defined the marital status criterion, because it affects the household tax situation and the documents required. Marriage or cohabitation may have tax implications and requires the submission of specific documents.

The ontology also includes user profiles, which are defined by a combination of profile elements. For example, a single person with real estate and a car would be defined by the combination:

Célibataire ⊓ AvecBienImmobilier ⊓ AvecBienMobilier

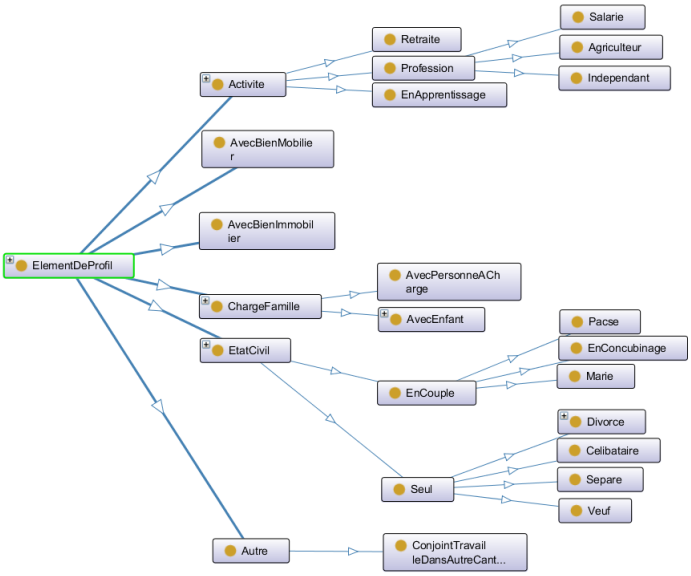


Figure 5: The section of *User Profiles* within the ontology (figure taken from Protégé ontology editor tool)

(iii) Tax Items. The ontology defines the tax items, which represent the different categories of taxes and duties applicable in a given Geneva tax jurisdiction. The list of these items is directly linked to the Geneva Tax return form. Each document in the ontology is associated with one or more tax items. Figure 6 provides a visual representation. Tax items are organised into three groups: Deduction (linked to any deduction we can provide such as doctors' bills or work related travel expenses); Fortune (savings accounts, real estate, etc.); Revenu (any type of income or rent from various activities).

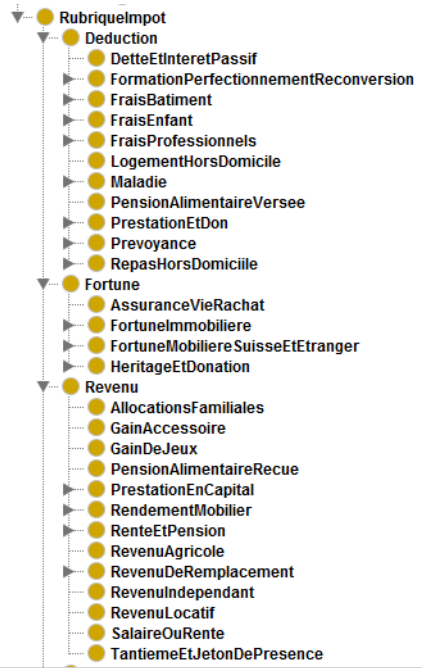


Figure 6: The section of *Tax Items* within the ontology

(iv) Changes. The ontology includes a definition for household profile changes. Figure 7 gives a visual representation of the section of changes. Similarly to the Tax items above, it is linked to the Geneva Tax return form and includes information related to changes in marital status, income activities, children, domicile, etc.

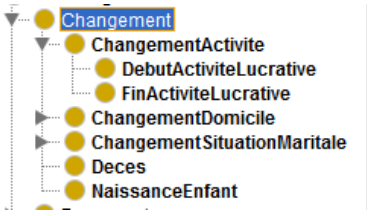


Figure 7: The section of *changes* within the ontology

3) **Defining the properties of the classes**
We defined properties for documents in relation to data that we need to extract from the documents and that serve to classify, label the document, or build a person or household profile.
For each class of document, these elements are identified through a combination of two methods: data extraction and evaluation of their usefulness for the documents in question. We identified the most recurrent properties for all documents, as shown in table 1. Specific machine learning algorithms are employed to extract information from administrative documents (e.g. from health insurance documents). This information is then stored as an RDF triple on which further reasoning will apply.

Table 1
Ontology Properties

Property	Description
nomPersonne	The name of a person
dateNaissance	The date of birth of a person
adressePersonne	The address of a person
emetteur	The issuer of an invoice
destinataire	The recipient of an invoice
echeanceContrat	The deadline for fulfilling the terms of a contract
canton	The canton in which a taxpayer is domiciled
commune	The municipality in which a taxpayer is domiciled
noAVS	The AVS number of a taxpayer
devise	The currency used in a fiscal document
noClient	The client number assigned to a taxpayer by a tax authority
noContrat	The unique identifier assigned to a contract
noCompte	The bank account number associated with a taxpayer
noDepot	The file number assigned to a tax file
codePostal	The postal code of a taxpayer's address
codeCommuneTravail	The municipality in which a taxpayer works
adresseLieuTravail	The address of a taxpayer's workplace
anneeFiscale	The fiscal year to which a tax document or liability relates
montant	The monetary amount associated with a fiscal document

Figure 8 shows the common properties for an account certificate, such as "IBAN", "account number", "user client", "account opening date" and "account closing date".

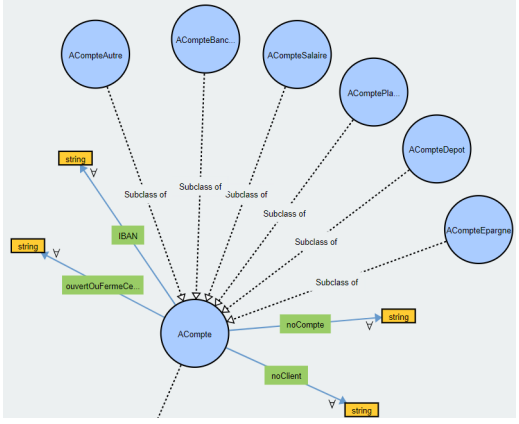


Figure 8: Ontology properties for an account certificate (figure adopted by WebVOWL tool)

5. Data Model expressed with Description Logic

In order to perform semantic reasoning, we need to define "shapes" to describe the structure, properties, and relationships of data. We first express here these shapes and relationships in Description Logic, before expressing them in SHACL in the next Section.

This approach addresses the research questions A, B, and C defined in Section 3.

A) **Document classification and multi-labelling rules**
Classes of documents are uniquely defined by distinctive characteristics. For example, a salary certificate document must contain the employee's last name and first name, the employer, and the amount of the salary. The following Description Logic (DL) notation represents some basic requirements that must be met for a document to be classified as a salary certificate.

SalaryCertificate
$\sqsubseteq delivered By$ some $Employee$
$\sqcap contains$ some $EmployeeLastName$
$\sqcap contains$ some $EmployeeFirstName$
$\sqcap contains$ some $SalaryAmount$

For the multi-labelling rule, one or more labels are assigned to each document to allow the automatic organisation of the documents into several predefined categories. The following DL notation is used to assign one or more labels to a salary statement that has a double "Tag".

SalaryCertificate
$\sqsubseteq hasTag$ some $Income$
$\sqsubseteq hasTag$ some Tax

B) **Customer profile rules**
In order to infer the users' profile by analysing the documents they provided, some **direct rules** are defined. For example, if a user delivers a salary certificate then the user is tagged as being an employee. The following DL notation allows the automated inference of user profile based on the documents the user provide.

DocumentDelivery
$\sqsubseteq delivered By$ some $User$
$\sqcap isType$ some $SalaryCertificate$
$\sqsubseteq deliverd By$ some $Employee$

C) **Documents delivery rules**
In order to infer which documents match the profile of the user, some **inverse rules** are defined. For example, if a person is tagged as an employee, then this person has to deliver a salary certificate. Additionally, each person has to provide a health insurance policy. The following DL notation allows for the automated inference of the type of documents to be delivered based on the user's profile.

$Person$
$\sqcap isType$ some $Employee$
$\sqsubseteq hasToDeliver$ some $SalaryCertificate$

The SHACL shapes are useful to define a set of property shapes and semantic rules for multi-label document classification and user profiling.
Specifically, we defined 92 property shapes and three sets of semantic rules, resulting in a total of 120 rules. These rules included 78 multi-label rules, 21 customer profile rules and 21 document delivery rules. Using SHACL allowed us to validate RDF data against these rules and ensure that the data complied with the defined constraints.

6. Implementation

We implemented SHACL node shapes for each class of the ontology. Each node shape is linked to its respective class through `sh:targetClass` property. By defining a target class within the shape's definition, it becomes applicable to all instances of that specific class. For example, *Salary Certificate Shape* is a type of node shape, and it is connected to the *SalaryCertifcate* class. By using the target, it is possible to ensure that all instances of *SalaryCertificate* are checked against the conditions defined within the *SalaryCertificateShape*. Below, we present the implementation of the SHACL shapes, following the adopted methodology described in section 5.

A) **Classification of documents**
For each document that a user should submit, we identify its *sine qua non* elements.
Within the relative SHACL shape, we defined the relevant elements the document must contain as SHACL property shapes. As shown in listing 1, a salary certificate has to contain: the employee's surname and first name, one and only one employee, and the amount.
The predicates have constraints that can describe different values for each attribute shape. We use pre-built constraint types as `sh:datatype` to describe the type of literal values; `sh:minCount` to describe the maximum required number of values; `sh:maxCount` to specify the maximum number of value nodes.
By using such a SHACL shape, we can run a validation process that validates (or not) the document as being of the appropriate type. This can also be interpreted as "if the document contains all *sine qua non* elements, i.e. the validation is positive, then it is of that specific class", and is therefore assigned to that class.

A) **Multi-labelling documents**
Multi-labelling rules assign one (or more) label to each document. The assigned labels can then be used to automatically organise documents into predefined categories. We defined such rules as SHACL inference rules and their execution generates inferred triples of the form:

```
< document impots:tag label >
```

The `document` is the RDF individual of the document that is being labelled; `impots:tag` is a data property, defined in the ontology, for assigning the label to a document; and `label` is a string literal (`xsd:string`) containing the actual text value of the label.
Listing 2 shows a rule labelling a document of type salary certificate as both a tax document and an income document.

```
impots:SalaryCertificateShape
  rdf:type sh:NodeShape ;
      sh:property [
      sh:path impots:personSurname ;
      sh:datatype xsd:string ;
      sh:minCount 1 ;
      sh:maxCount 1 ;
      sh:class impots:PersonSurname ;
      sh:name "Person Surname" ;
    ] ;
    sh:property [
      sh:path impots:employer ;
      sh:minCount 1 ;
      sh:maxCount 1 ;
      sh:class impots:Employer ;
      sh:name "Employer" ;
    ] ;
    sh:property [
      sh:path impots:personFirstName ;
      sh:datatype xsd:string ;
      sh:minCount 1 ;
      sh:maxCount 1 ;
      sh:class impots:PersonFirstName ;
      sh:name "Person First Name" ;
    ] ;
  sh:property [
      sh:path impots:amount ;
      sh:datatype xsd:string ;
      sh:minCount 1 ;
      sh:maxCount 1 ;
      sh:class impots:SalaryAmount ;
      sh:name "Salary Amount" ;
    ] ;
.
```

Listing 1: Relevant features of a salary certificate document represented as SHACL shapes

```
impots:SalaryCertificateShape
    rdf:type sh:NodeShape ;
    sh:rule [
        rdf:type sh:TripleRule ;
        sh:subject sh:this ;
        sh:predicate impots:tag ;
        sh:object "Tax" ;
    ] ;
    sh:rule [
        rdf:type sh:TripleRule ;
        sh:subject sh:this ;
        sh:predicate impots:tag ;
        sh:object "Income" ;
    ] ;
    sh:targetClass impots:SalaryCertificate ;
.
```

Listing 2: SHACL inference rule for labelling a document of type salary certificate

Suppose we have a knowledge graph containing data from various documents, including some salary certificates. By executing the rule shown in listing 2, we can infer new triples that explicitly state the type of the salary certificate. These inferred triples are shown in listing 3.

```
:documentX impots:tag "Tax" .
:documentX impots:tag "Income" .
```

Listing 3: Triples inferred by the inference rule shown in listing 2

B) **Customer profile rules**

As multi-labelling rules, the customers' profile rules are defined as SHACL inference rules.

Listing 4 shows an example of such rules. Contrary to the example previously shown (where the targeted documents are all the RDF individuals of a defined class), this example shows an extended targeting condition expressed using the SPARQL language.

The rule defined in listing 4 states that all nodes that satisfy the shape must have the `rdf:type property` equal to `impots:Employee`.

Therefore, if all nodes that represent the recipients of a salary certificate have `rdf:type` equal to employee, the validation of the shape will be successful. On the other hand, if one or more recipients of a salary certificate are not correctly represented in the RDF graph (because `rdf:type` is different from employee), the validation of the shape will be negative and an error will be reported.

```
impots:SalaryCertificate_Employee-Shape
    rdf:type sh:NodeShape ;
    sh:rule [
        rdf:type sh:TripleRule ;
        sh:object impots:Employee ;
        sh:predicate rdf:type ;
        sh:subject sh:this ;
    ] ;
    sh:target [
        rdf:type sh:SPARQLTarget ;
        sh:prefixes impots: rdf: ;
        sh:select """
            SELECT ?this
            WHERE {
                ?sc rdf:type
                    impots:SalaryCertificate .
                ?sc impots:recipient ?this .
                ?this rdf:type impots:Person .
            }
        """ ;
    ] ;
.
```

Listing 4: SHACL direct rule inferring the employee profile of a person from the salary certificate provided

The execution of this direct rule infers new RDF triples of the form:

`< person rdf:type impots:Employee >`

where `person` corresponds to the specific RDF individual; `rdf:type` is the property used to state that a resource is an instance of a class; and `impots:Employee` is the inferred class to which `person` belongs.

C) **Document delivery rules**

The rule defined within `impots:EmployeeShape` in listing 5 aims to verify the correspondence between employees and the salary certificates they have delivered. In particular, the shape uses a validation rule that requires all nodes representing employees to have a "deliver" relationship with at least one node representing a salary certificate.

```
impots:EmployeeShape
  rdf:type sh:NodeShape ;
  sh:rule [
      rdf:type sh:TripleRule ;
      sh:subject sh:this ;
      sh:predicate impots:delivers ;
      sh:object impots:SalaryCertificate;
    ] ;
  sh:targetClass impots:Employee ;
.
```

Listing 5: SHACL inverse rule inferring the need for an employee profile to deliver a salary certificate

The execution of the rule defined in listing 5 infers new triples of the form:

```
< employee impots:delivers impots:SalaryCertificate >
```

which means that every employee has to deliver a salary certificate.
The shape of the rule *PersonShape* in listing 6 uses a validation rule, expressed as a triple rule, which specifies that all nodes representing persons must have a deliver relationship with at least one node representing an instance of health insurance. In other words, the shape verifies whether the persons have delivered at least one health insurance certificate.

```
impots:PersonShape
  rdf:type sh:NodeShape ;
  sh:rule [
      rdf:type sh:TripleRule ;
      sh:subject sh:this ;
      sh:predicate impots:delivers ;
       sh:object impots:HealthInsurance;
    ] ;
  sh:targetClass impots:Person ;
.
```

Listing 6: SHACL inverse rule for inferring that a person profile needs to deliver a Health Insurance

The execution of the rule defined in listing 6 infers new triples of the form:

```
< person impots:delivers impots:HealthInsurance >
```

which in turn means that each person has to deliver a health insurance policy document.

7. Validating and Evaluation

We first show how we validate the data against SHACL shapes, and second how we evaluate the rules.

7.1. Validating data with SHACL Shape

To validate the RDF data against the defined SHACL shapes, we use a SHACL validation engine such as the one integrated into TopBraid Composer[2], called the TopBraid Validator.

We create SHACL validation test cases in TopBraid Composer to ensure that the RDF data conforms to the specified shapes. These test cases define a set of RDF data and corresponding SHACL shapes, as well as validation constraints that must be applied to this data to verify its compliance with the RDF data model specifications. The test cases allow the correct implementation of the SHACL validation rules to be verified and any validation errors to be detected. This ensures that the data is accurate, consistent and conforms to the specifications of the RDF data model.

We wrote six graph validation test cases with regard to 3rd pillar attestation, Deposit account attestation, AVS[3] pension or disability insurance attestation, LPP[4] pension attestation, and Salary certificate. Each test case performs a SHACL constraint validation on the entire graph and compares the results with the expected validation results stored with the test case.

The test case for "PersonShape", in listing 7, defines two instances of the `impots:Person` class: an invalid resource and a valid resource. The former has a value for the *noAVS* property that violates a constraint defined in the *AVS shape*. Indeed, the AVS number must always start with "756", and must be followed by two groups of 4 digits, and finish with a group of two digits. In this case, it properly starts with "756", but then continues with only three digits "023" instead of four. The latter satisfies all the constraints defined in the shape, as the second group is made of 4 digits and is "0123". We deliberately insert this error to verify that the SHACL rules work correctly and are able to detect any problems or violations of the specified constraints.

The expected result of the validation is defined as a validation report, with information about the constraint that was violated, the form that defines the constraint, the path to the property that caused the violation, the value causing the violation, and the severity of the violation. The report will also indicate that the validation did not conform.

[2]For Top Braid Composer, see: http://www.topquadrant.com/products/TB_Composer.html

[3]The AVS or OASI number is the social insurance number uniquely associated with individuals in Switzerland https://www.bsv.admin.ch/bsv/en/home/social-insurance/ahv/legal-bases-and-legislation/ahv-nummer.html

[4]Pension related fund

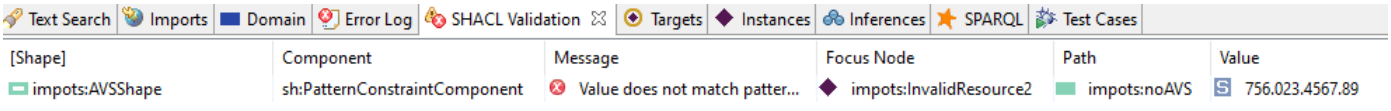


Figure 9: Results of SHACL validation

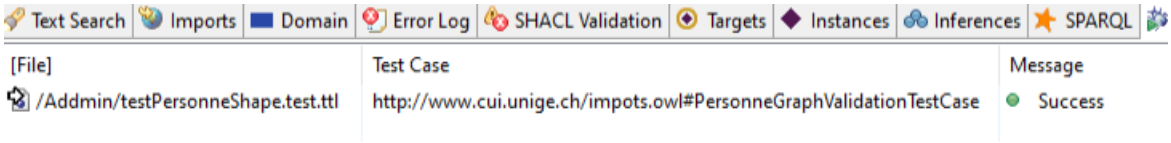


Figure 10: Result of "Person Shape" test case

```
<http://www.cui.unige.ch/PersonShape.test>
  rdf:type owl:Ontology ;
  rdfs:label "Test of PersonShape" ;
  owl:imports <http://datashapes.org/dash> ;
  owl:imports <http://www.cui.unige.ch/impots.shapes> ;
  owl:versionInfo "Created with TopBraid Composer" ;
.
impots:InvalidResource2
  rdf:type impots:Person ;
  impots:noAVS "756.023.4567.89" ;
  impots:personSurname "Zola" ;
  impots:personFirstName "Giovanna" ;
.
impots:PersonGraphValidationTestCase
  rdf:type dash:GraphValidationTestCase ;
  dash:expectedResult [
      rdf:type sh:ValidationReport ;
      sh:conforms "false"^^xsd:boolean ;
      sh:result [
          rdf:type sh:ValidationResult ;
          sh:focusNode impots:InvalidResource2 ;
          sh:resultPath impots:noAVS ;
          sh:resultSeverity sh:Violation ;
          sh:sourceConstraintComponent
          sh:PatternConstraintComponent ;
          sh:sourceShape impots:AVSShape ;
          sh:value "756.023.4567.89" ;
        ] ;
      .
impots:ValidResource
  rdf:type impots:Person ;
  impots:noAVS "756.0123.4567.89" ;
  impots:personSurname "Zola" ;
  impots:personFirstName "Giovanna" ;
.
```

Listing 7: Graph validation test case of "PersonShape"

The SHACL test case returned the error message for the invalid resource *AVS*, as shown in figure 9. The presence of this error message indicates that the SHACL rules are working correctly and that the invalid resource has been identified and reported.

Figure 10 shows that the result of the "PersonShape" test case has been successful. This means that all the data instances that satisfy the "PersonShape" also satisfy the SHACL rules specified for that shape. This is a positive result, indicating that the validated data conforms to the SHACL rules and that the applied SHACL rules work correctly on this data.

Figure 11 shows positive results for the other five tests cases concerning five tax documents, such as 3rd pillar attestation, deposit account attestation, AVS pension or disability insurance attestation, LPP pension attestation, salary certificate.

7.2. Evaluation

After validating the data against the defined shapes, we applied SHACL rules to the dataset to generate new information and improve data quality. In this paragraph, we will describe the results of applying the SHACL rules on the validated data and evaluate the effectiveness of the SHACL rules in meeting the requirements of the application.

7.2.1. Evaluation of multi-labelling rules

As we can see in figure 12, the execution of the rules shown in listing 2 infers two new triples that assign the two labels "Income" and "Tax" to the document with ID "Salary C 12.3.334", which is of type "SalaryCertificate".

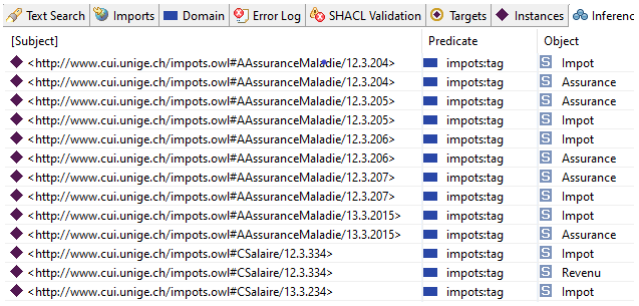


Figure 12: Inferred triples that assign two labels to a document of type salary certificate

7.2.2. Evaluation of users profile rules

We defined two individuals Zola Giovanna and Ladoumegue Jules. We assume that Zola Giovanna has provided a salary certificate. Based on the direct rule defined in listing 4, since Zola Giovanna delivered such a certificate, the rule infers that she is an employee. Figure 13 shows the inferred triples.

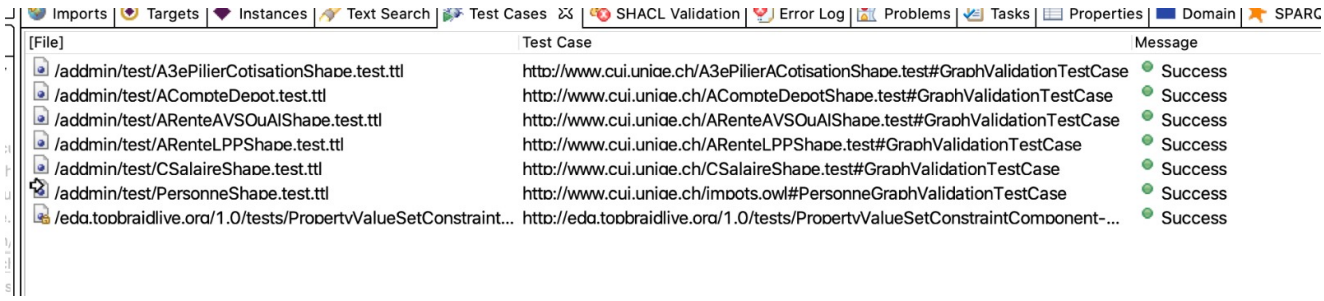


Figure 11: Result of six shape test cases

Figure 13: The result of the execution of direct rules to a person who has issued a salary certificate

Conversely, we defined Ladoumegue Jules as an employee. Therefore, according to the inverse rule defined in listing 5, the execution infers that since he is of class employee, he must deliver a salary certificate document. According to Listing 6, since he is also a person, he has to provide a health insurance policy. Figure 14 shows the inferences mentioned.

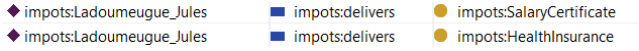


Figure 14: The result of running inverse rules on a person with an employee profile

8. Conclusion

This paper presents a semantic rule-based approach for a semantically enriched DMS, which facilitates the management of administrative documents, user profiling, and other related document management activities. This approach is capable of overcoming the limitations of traditional Document Management Systems (DMS) that rely solely on metadata to organise documents. The proposed approach uses a combination of ontology and SHACL rules to capture knowledge of the domain and legal regulations, validate data, and infer new information. The process is designed to be dynamic and based on data provided by users. Additionally, the process takes into account users' profiles and underlying rules to enable accurate and automated document management.

The paper demonstrates the innovative nature of the proposed approach and its potential to improve the accuracy and completeness of information managed by a DMS. The ontology used in the process captures specific concepts of Swiss tax returns, while the SHACL rules are used to validate and reason about asserted RDF triples of actual data from different tax households on the basis of actual regulations.

The limited availability of Swiss tax documents presented a major challenge during the development of the system. As a large dataset of tax documents is required to perform machine learning tasks and related analyses, this was an obstacle to the progress of the project. This had an impact on the quality of the information extraction module. However, it did not interfere with the ontology or the SHACL rules, which cover most of the various household configurations. Nevertheless, the implementation still needs to be validated on a large dataset of documents. Future work will consider the dynamic nature of profiles and the integration of such a module into a wider DMS service.

Furthermore, two additional challenges were identified during the development of the project. The first challenge was dealing with multilingualism, as Switzerland has four official languages (German, French, Italian, and Romansh). This required the project team to develop techniques for processing and analysing tax documents in multiple languages. The second challenge was managing the different tax laws enacted by the cantons. Each canton has its own laws, rules, tax documents, and procedures, so it was necessary to develop a flexible system that could adapt to the specific requirements of each canton.

It should be noted that while the project has a broader scope, targeting any type of administrative document, the work presented in this paper focuses specifically on Swiss tax return documents written in French. The project has not dealt with German, Italian, or English documents.

It is important to note that a business-to-business (B2B) system for professionals integrating this solution must also take into account privacy and confidentiality issues. These concerns must be carefully considered and addressed to ensure that the system complies with relevant laws and regulations regarding privacy and confidentiality.

In general, the adoption of our semantic approach could simplify the management of administrative documents and improve user profiling. The proposed DMS can be used in various contexts, such as tax filing, document management for an insurance company, or legal document management.

Acknowledgments

This research was supported by Innosuisse within the framework of the innovation project 50606.1 IP-ICT "Addmin". The authors thank Anne-Françoise Cutting-Decelle, Assane Wade, Claudine Métral, Gilles Falquet, Graham Cutting, and Sami Ghadfi for their valuable collaboration with

the “Addmin” project.

References

[1] Abbasova, V., 2020. Main concepts of the document management system required for its implementation in enterprises. ScienceRise 1, 32–37. doi:10.21303/sr.v0i1.1149.

[2] Amato, F., Casola, V., Mazzocca, N., Romano, S., 2011. A semantic-based document processing framework: A security perspective, in: 2011 International Conference on Complex, Intelligent, and Software Intensive Systems, pp. 197–202. doi:10.1109/CISIS.2011.37.

[3] Cappelli, M.A., Caselli, A., Di Marzo Serugendo, G., 2023. Enriching rdf-based document management system with semantic-based reasoning, in: Chang, S. (Ed.), The 29th International DMS Conference on Visualization and Visual Languages, KSIR Virtual Conference Center, USA, June 29-July 3, 2023, KSI Research Inc.. pp. 44–50. URL: https://doi.org/10.18293/DMSVIVA23-034, doi:10.18293/DMSVIVA23-034.

[4] Di Marzo Serugendo, G., Falquet, G., Metral, C., Cappelli, M.A., Wade, A., Ghadfi, S., Cutting-Decelle, A.F., Caselli, A., Cutting, G., 2022. Addmin: Private computing for consumers’ online documents access: Scientific technical report .

[5] Fuertes, A., Forcada, N., Casals, M., Gangolells, M., Roca, X., 2007. Development of an ontology for the document management systems for construction, in: Complex Systems Concurrent Engineering. Springer, pp. 529–536.

[6] Gorelashvili, L., 2023. The importance of digitalization of legal documents preparing process and its impact on peoples’ legal guarantees, in: Geibel, R., Machavariani, S. (Eds.), Digital Management in Covid-19 Pandemic and Post-Pandemic Times. Springer, Cham. doi:10.1007/978-3-031-20148-6_3.

[7] Gostojić, S., Sladić, G., Milosavljević, B., Zarić, M., Konjović, Z., 2014. Semantic driven document and workflow management, in: Proceedings of the international conference on applied internet and information technologies (ICAIIT 2014). Zrenjanin, Serbia, pp. 229–234.

[8] IEC, I., 2001. 82045-1, document management–part 1: Principles and methods. International Organization for Standardization .

[9] Knublauch, H., Kontokostas, D., 2017. Shapes Constraint Language (SHACL). W3C Recommendation. W3C. URL: https://www.w3.org/TR/2017/REC-shacl-20170720/.

[10] Lee, Y.H., Hu, P.J.H., Tsao, W.J., Li, L., 2021. Use of a domain-specific ontology to support automated document categorization at the concept level: Method development and evaluation. Expert Systems with Applications 174, 114681. URL: https://www.sciencedirect.com/science/article/pii/S0957417421001226, doi:https://doi.org/10.1016/j.eswa.2021.114681.

[11] Leukel, J., Schuele, M., Scheuermann, A., Ressel, D., Kessler, W., 2011. Cooperative semantic document management, in: Business Information Systems: 14th International Conference, BIS 2011, Poznań, Poland, June 15-17, 2011. Proceedings 14, Springer. pp. 254–265.

[12] Musen, M.A., 2015. The protégé project: a look back and a look forward. AI Matters 1, 4–12. URL: https://doi.org/10.1145/2757001.2757003, doi:10.1145/2757001.2757003.

[13] Noy, N.F., McGuinness, D.L., et al., 2001. Ontology development 101: A guide to creating your first ontology.

[14] Sheng, L., Lingling, L., 2011. Application of ontology in e-government, in: 2011 Fifth International Conference on Management of e-Commerce and e-Government, IEEE. pp. 93–96.

[15] Sladić, G., Cverdelj-Fogaraši, I., Gostojić, S., Savić, G., Segedinac, M., Zarić, M., 2017. Multilayer document model for semantic document management services. Journal of Documentation 73, 803–824.

[16] Stylianou, N., Vlachava, D., Konstantinidis, I., Bassiliades, N., Peristeras, V., 2022. Doc2kg: Transforming document repositories to knowledge graphs. Int. J. Semantic Web Inf. Syst. 18, 1–20. URL: https://doi.org/10.4018/ijswis.295552, doi:10.4018/ijswis.295552.

[17] Uschold, M., Gruninger, M., 1996. Ontologies: Principles, methods and applications. The knowledge engineering review 11, 93–136.

[18] Wang, G., Wang, B., Han, D., Qiao, B., 2005. Design and implementation of a semantic document management system. Information Technology Journal 4, 21–31.

[19] Yousufi, M., 2023. Exploring paperless working: A step towards low carbon footprint. European Journal of Sustainable Development Research 7.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

A Semantic Comparative Analysis of Agile Teamwork Quality Instruments in Agile Software Development[⋆,⋆⋆]

Ramon Santos[a], Felipe Cunha[a], Thiago Rique[a], Mirko Perkusich[a], Ademar Neto[a], Danyllo Albuquerque[a], Hyggo Almeida[a] and Angelo Perkusich[a]

[a]Federal University of Campina Grande, Paraiba, Brazil

ARTICLE INFO

Article History:
Submitted 5.10.2023
Revised 8.7.2023
Accepted 8.8.2023

Keywords:
agile software development
teamwork quality
teamwork effectiveness
teamwork quality instruments
semantic comparative analysis

ABSTRACT

Multiple models (or instruments) for measuring Teamwork Quality (TWQ) and Teamwork Effectiveness (TWE) for Agile Software Development (ASD) have been created. Regardless, such models have different constructs and measures, with a limited understanding of how they are semantically related. [Objective] Our goal is to understand how specific instruments for ASD are related, considering the semantic relationship between them. [Method] We analyzed eight specific teamwork instruments for ASD (ASD instruments), comparing quantitative factors to identify which such instruments use most. Then, we compared them qualitatively from a semantic perspective, given that they are specific instruments in an agile context, considering the solid theories that support them. [Results] The results showed that Team Orientation and Coordination were identified among the top three rankings, both in the frequency of instrument questions and in the frequencies of literature-based Thematic Network themes. We found in our semantic analysis important themes associated a many instrument factors: Team Interaction associated with Communication factor, Acceptance of Goals associated with Coordination, etc. Qualitative concepts can be investigated considering the ASD factors from the knowledge of the identified parts of the agile instruments. [Conclusion] The semantic analysis brings new perspectives for researchers and practitioners to highlight more investigation about different teamwork aspects (new instruments themes) in ASD. We argue the need to add other ASD instruments to be compared to solidify the results found in this study, so we advocate further studies on this topic.

1. Introduction

The success of Agile Software Development (ASD) heavily relies on the competencies, interactions, and skills of its professionals [27, 31]. As software teams are the critical source of agility in ASD [32, 10], people are a crucial resource [23, 32, 3], and the quality of team interactions can significantly impact a project's outcome. Hence, Teamwork Quality (TWQ) is essential for agile projects' success [15, 6, 16]. The industry is rapidly adopting ASD [29], and the need for systematic team development [22] has compelled researchers to focus on teamwork aspects increasingly.

A team can be defined as a social system of two or more people which is embedded in an organization (context) whose members perceive themselves as such and are perceived as members by others (identity), collaborating on a common task (teamwork) [1, 12, 11]. The main focus of TWQ research is on the quality of interactions within teams rather than team members' (task) activities. Starting from the widespread fundamental proposition that the success of work conducted in teams depends (beyond the quantity and correctness of the task activities) on how well team members collaborate or interact.

The construct TWQ was proposed [13] as a comprehensive concept of the quality of team interactions. To capture the nature of team members working together, six facets of

ramon.santos@virtus.ufcg.edu.br (R. Santos); felipe.cunha@virtus.ufcg.edu.br (F. Cunha); thiago.rique@virtus.ufcg.edu.br (T. Rique); mirko@virtus.ufcg.edu.br (M. Perkusich); ademar.sousa@virtus.ufcg.edu.br (A. Neto); danyllo.albuquerque@virtus.ufcg.edu.br (D. Albuquerque); hyggo@virtus.ufcg.edu.br (H. Almeida); angelo.perkusich@virtus.ufcg.edu.br (A. Perkusich)

ORCID(s): 0000-0003-4864-0480 (R. Santos); 0000-0003-4864-0480 (F. Cunha); 0000-0003-0897-4953 (T. Rique); 0000-0002-9433-4962 (M. Perkusich); 0000-0002-1651-4159 (A. Neto); 0000-0001-5515-7812 (D. Albuquerque); 0000-0002-2808-8169 (H. Almeida); 0000-0002-7377-1258 (A. Perkusich)

DOI reference number: 10-18293/JVLC2023-N2-036

the collaborative team process integrate into the concept of TWQ: Communication, Coordination, Balance of Member Contribution, Mutual Support, Effort, and Cohesion. These facets capture both task-related and social interaction within teams. Research has shown that TWQ has a positive impact on team development [13], increasing the chances of succeeding with ASD. [13][20][22].

In this context, researchers have proposed instruments for assessing teamwork quality in the agile context, such as (i) a Radar Plot [21] that considers five dimensions for assessing TWQ: Shared Leadership, Orientation, Redundancy, Learning, and Autonomy; (ii) a Structural Equation Model [15] (TWQ-SEM), based on a differentiated replication from [13], which considered that the teamwork construct is comprised of six variables: Communication, Coordination, Balance of Member Contribution, Mutual Support, Effort, and Cohesion.

All the instruments mentioned are generic and cannot represent specific situations in the agile context. This was evidenced by the emergence of new instruments tailored for Agile Software Development (ASD). For instance, the aTWQ instrument [22] was developed based on the TWQ instrument [13], while the ATEM instrument [30] was developed based on the Big Five theory [25]. Additionally, a Bayesian networks-based model (TWQ-BN) [8] was developed based on the TWQ instrument [13]. Moreover, the TACT instrument [9] was developed based on the TCI instrument [2], and finally, the STEM instrument [33] was developed considering that some specific factors in Scrum.

Although the literature on TWQ has evolved, there was no unified understanding of what factors influence teamwork in ASD. Silva et al. [28] took a first step toward better understanding the relationship between agile TWQ instruments by performing a quantitative comparison between TWQ-SEM [15] and TWQ-BN [8]. However, the study is limited to only two instruments and focused on a high-level analysis (i.e., factors), not explicitly considering the instruments‘ questions.

Freire et al. [7] took a step further by developing a literature-based Thematic Network identifying the most frequent codes and themes in agile teamwork literature. Freire et al. [7] argued that researchers and practitioners can use the thematic network as a reference for understanding the factors and dimensions that comprise ASD Teamwork. With this, practitioners can, for example, define mechanisms to monitor such dimensions and use the collected data as a reference to drive actions toward improving the team's performance.

In our earlier research [26], we used Freire et al. [7]'s thematic network as a reference for analyzing three ASD teamwork instruments: ATEM, aTWQ, and TWQ-BN. However, we only performed a syntactic (i.e., quantitative) analysis, which brings many limitations, such as loss of information. This study complements our past research by considering eight ASD teamwork instruments and performing a semantic (i.e., qualitative) analysis. This paper provides a more comprehensive understanding of the interrelationships between factors and questions within the instruments, enhancing comprehension of their functioning.

Noteworthy enhancements and novel contributions in this paper, not covered in Santos et al. [26], include the following:

- **Expanded Scope of Comparison**: The quantity of compared ASD instruments has been increased to eight, all of which were identified in our systematic literature review (SLR) work, soon to be published in the 37th Brazilian Symposium on Software Engineering (SBES 2023).
- **Enhanced ASD Instruments Factors and Freire et al. [7]** Themes Comparison: The comparison now encompasses eight instruments, leading to more robust and dependable results.
- **Semantic Comparison**: A refined approach has been adopted for comparing instruments' questions based on a semantic analysis of their factors and questions.
- **Investigation of Teamwork Instruments Factors Evolution**: The association between the chronological evolution of the instruments and the evolution of subjects associated with the factors of these instruments has been thoroughly investigated, revealing discernible patterns and trends.

This paper is organized as follows: Section 2 presents teamwork theoretical concepts and general information on the ASD Teamwork instruments compared in this work. Section 3 describes the research questions. Section 4 presents the quantitative comparison. Section 5 presents the instruments' semantic comparison. Section 6 presents the Discussion of the results. Section 7 covers the study's limitations and threats to validity. Section 8 presents the study implications. Lastly, Section 9 presents our final remarks, discussing potential future work.

2. Background

The topic of TWQ assessment has garnered considerable attention in the ASD research community [8, 22, 9, 30, 33]. This section provides an overview of the main concepts related to this field of research relevant to our study. Section 2.1 defines what is a "teamwork instrument" in the scope of our research and elucidates the distinction between the concepts of "Team effectiveness" and "Team performance." Secondly, Section 2.2 presents a comprehensive overview of the eight ASD teamwork instruments objects of our study. We identified such instruments through a Systematic Literature Review (In press). Lastly, Section 2.3 discusses the theoretical evolution of ASD teamwork instruments.

2.1. Teamwork Models

This section defines what is "teamwork instrument" in the scope of our research and discusses fundamental concepts of teamwork models in software engineering.

Definition of a "teamwork instrument". A teamwork instrument is an assessment tool designed to capture and evaluate various factors pertaining to teamwork. Typically, it comprises questions or statements specifically crafted to gather information and assess specific aspects of team collaboration, communication, coordination, and other relevant

Table 1
Teamwork Quality Instruments used in Agile Software Development.

Instrument Number	Year	Title	Instrument
I1	2001	Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence	**TWQ** - Teamwork Quality
I2	2009	Putting Agile Teamwork to the Test – An Preliminary Instrument for Empirically Assessing and Improving Agile Software Development	**Radar Plot**
I3	2010	A teamwork model for understanding an agile team: A case study of a Scrum project	**ASTM-Agile Scrum Teamwork Model**
I4	2018	A Bayesian networks-based approach to assess and improve the teamwork quality of agile teams	**TWQ-BN** - Teamwork Quality - Bayesian network
I5	2020	Evaluation of Agile Team Work Quality	**aTWQ** - Agile Teamwork Quality
I6	2020	An Instrument to Assess the Organizational Climate of Agile Teams - A Preliminary Study	**TACT** - Assess the Organizational Climate of Agile Teams
I7	2022	A teamwork effectiveness model for agile software development	**ATEM** - Agile Team Effectiveness Model
I8	2022	A Theory of Scrum Team Effectiveness	**STEM** - Scrum Team Effectiveness Model

dimensions. Through administering such instruments, researchers or practitioners can systematically measure and evaluate different facets of teamwork, identify potential issues or barriers, and make informed decisions to enhance team performance and productivity. The prevalent technique employed in constructing these instruments is Structural Equation Modelling (SEM) [24], a large sample technique where a sample size of at least 200 is preferable [14]. For further guidance on building a teamwork instrument, readers can refer to the work of Marsicano et al. [17].

Team effectiveness x Team performance. The distinction between Team effectiveness and Team performance is highlighted in the work of Salas et al. [25]. Team performance is characterized as the outcome of a team's actions, irrespective of the approach employed to complete their task. In the context of software development, team performance encompasses meeting project goals, adhering to budgets and schedules, and delivering high-quality software. On the other hand, Team effectiveness is defined in a more comprehensive manner, encompassing how the team collaborates and interacts while accomplishing their tasks. This includes various team interactions, such as planning meetings, reviews, retrospectives, pair programming, and the use of coordination artifacts like iteration and product backlogs. In essence, team effectiveness considers not only the end result but also the dynamics and cooperation displayed during the task execution.

Team effectiveness models find frequent application in software engineering studies. Examples of such models include the Big Five model [25], which is utilized in various studies such as [6, 20, 30]), the Teamwork Quality model [13], featured in studies like [15, 22], and the Input-Process-Output (IPO) model [18], which is employed in studies like [19]. A comprehensive overview of these three models can be found in the work of Strode et al. [30].

In this work, we considered the TWQ instrument [13] as a comparative base because it has been extensively referenced in ASD [13, 8, 22, 15]. Also, we recognize that "teamwork quality" and "teamwork effectiveness" are closely related concepts that are commonly evaluated through measurable results [13, 33, 30, 22, 7]. Therefore, we refer to these concepts as "teamwork quality" or simply "teamwork".

2.2. Teamwork Instruments in ASD

This section summarizes the eight ASD teamwork instruments under study: TWQ instrument [13] (I1), Radar Plot instrument [21] (I2), ASTM [20] instrument (I3), TWQ-BN [8] instrument (I4), aTWQ) [22] instrument (I5), TACT [9] instrument (I6), ATEM [30] instrument (I7), and STEM [33] instrument (I8). Table 1 showcases a comprehensive list of all the teamwork instruments, along with the associated articles and their respective creation years.

TWQ - Teamwork Quality instrument (2001) [13]: Hoegl and Gemuenden [13] presented a comprehensive concept of collaboration in teams called Teamwork Quality (TWQ). This construct has six facets (i.e., Communication, Coordination, Balance of Member Contributions, Mutual Support, Effort, and Cohesion). Based on these facets and data collected in their study, the authors proposed a way for measuring the TWQ where the high order factor (i.e., TWQ) is the dependent variable, and the construct facets are the independent variable.

Radar Plot instrument (2009) [21]: Moe et al. [21] proposed an instrument that addresses key concerns and characteristics of agile teamwork and presents them along five dimensions: Shared Leadership, Team Orientation, Redundancy, Learning, and Autonomy. The instrument outputs a radar plot of the teamwork's status. To assess the teamwork's current status, it is necessary to answer a set of questions for each dimension and, based on these answers, assign a score on a scale from 0 to 10 for the dimension.

ASTM - A teamwork model for understanding an agile team: A case study of a Scrum project instrument (2010) [20]: Based on Dickinson and McIntyre's [4] teamwork model, Moe et al. [20] focused on the interrelations between essential teamwork components. Problems with team orientation, team leadership, and coordination, in addition to highly specialized skills and corresponding division of work, were important barriers to achieving team effectiveness.

TWQ-BN - Teamwork Quality Bayesian networks (2018) [8]: According to the agile principles and values, teamwork factors are critical to achieving success in agile projects. The TWQ-BN has a predicting and diagnosis purpose using Bayesian Networks. According to agile principles and values, teamwork factors are critical to achieving success in agile projects.

However, teamwork does not automatically arise. There are some existing instruments with the purpose of assessing the teamwork quality based on Structural Equation Modeling (i.e., empirically derived) and Radar Plot [21]. TWQ-BN instrument has 17 questions.

aTWQ - Agile Team Work Quality (2020) [22]: Based on Hoegl and Gemuenden's study [13] and a systematic literature review about challenges and success factors for large-scale agile transformations performed by Paasivaara et al. [5]. Poth et al. [22] derived the aTWQ at the initial team-level approach covering the following six factors: communication, coordination, balance of contribution, mutual support, effort, and cohesion. These six quality aspects lead to team performance [15], legitimating economically the effort for measurement and further TWQ improvement. They combined these aspects with those of TCI [2] and defined 19 related questions to develop a holistic team evaluation questionnaire for aTWQ [22].

TACT - An insTrument to Assess the organizational ClimaTe of agile teams (2020) [9]: TACT allowed for classifying the organizational climate of teams into the Communication, Collaboration, Leadership, Autonomy, Decision-Making, and Client Involvement dimensions. Some items were assessed negatively or neutrally, which represents points of attention. TACT captured the lack of agile ceremonies, the difficulty of the product owner in planning iterations, and the distance in leadership.

ATEM - Agile teamwork effectiveness model (2022) [30]: Teamwork is crucial in software development, particularly in agile development teams which are cross-functional and where team members work intensively together to develop a cohesive software solution. Effective teamwork is not easy; prior studies indicate challenges with communication, learning, prioritization, and leadership. Nevertheless, much advice is available for teams, from agile methods, practitioner literature, and general studies on teamwork to a growing body of empirical studies on teamwork in the specific context of ASD. The ATEM [30] model is based on evidence from focus groups, case studies, and multi-vocal literature and is a revision of a general Big Five [25] team effectiveness model. The ATEM [30] model comprises shared leadership, team mentoring, redundancy, adaptability, and peer feedback. Coordination mechanisms are needed to facilitate these components. Coordination mechanisms are shared mental models, communication, and mutual trust. ATEM instrument has 31 questions.

STEM - A Theory of Scrum Team Effectiveness Model (2022) [33]: The STEM model [33] proposes that the effectiveness of Scrum teams depends on five high-level factors - responsiveness, stakeholder concern, continuous improvement, team autonomy, and management support - and thirteen lower-level factors. The main finding is the interplay between stakeholder concern and responsiveness as drivers of agile team effectiveness. In turn, this requires a high degree of team autonomy, continuous improvement, and support from management.

2.3. Theoretical Evolution of Teamwork Instruments in ASD

Since the emergence of the TWQ instrument [13] in 2001, several other instruments have emerged in the literature. In this context, researchers have proposed instruments for assessing TWQ in the agile context, such as (i) a Radar Plot [21] that considers five dimensions for assessing TWQ (Shared Leadership, Orientation, Redundancy, Learning, and Autonomy); (ii) Moe et al. [20] used ASTM [20] that considers seven factors: Team orientation, Team leadership, Monitoring, Feedback, Backup, Coordination, and, Communication, (iii) a Structural Equation Model [15] (TWQ-SEM), based on a differentiated replication from Hoegl et al. [13], which considered that the teamwork construct is comprised of six variables: Communication, Coordination, Balance of Member Contribution, Mutual Support, Effort, and Cohesion. All the instruments mentioned are generic and cannot represent specific situations in the agile context because the instrument questions are not focused on agile terms.

Since 2018, specific instruments for ASD have emerged: a Bayesian networks-based model (TWQ-BN) [8] was developed based on the TWQ [13] instrument. The aTWQ instrument [22] was developed based on the TWQ [13] instrument. The ATEM instrument [30] was developed based on the Big Five theory [25]. The TACT instrument [9] was developed based on the TCI instrument [2]. The ATEM instrument [30] was developed to measure team effectiveness in the agile context. The STEM instrument [33] was developed considering some specific factors in Scrum. All the mentioned instruments have something in common: they have instrument questions directly associated with agile context situations. In this work, we named these instruments Specific Agile Teamwork Instruments because they are specific for ASD.

Based on this observation, we propose the classification of agile teamwork instruments into two groups: Generic teamwork instruments and Agile-based teamwork instruments. The generic ones were developed until 2018: TWQ, Radar Plot, and ASTM. The Agile-based ones were developed in 2018: TWQ-BN, aTWQ, TACT, ATEM, and STEM.

We found that, generally, the instruments are built and supported by a general theory in literature. Observing this, we created a *Level 1* in this architecture (Figure 1). As examples, we can cite the theories in Teamwork Literature: the Teamwork Quality Theory [13], the Team Climate Theory [2], the Big Five Theory [25] and The Group Development Theory [34]. In *Level 2*, the theories are combined with empirical research to build the instruments, as examples we have: the TWQ instrument [13], the TCI instrument [2], the GDQ instrument [34], the ATEM instrument [30], and the STEM instrument [33]. In *Level 3*, the theories and instruments are combined to build new specific ones. In the case of our study, for ASD, as examples we have: the TWQ-BN instrument [8], built taking as reference the TWQ instrument [13], the aTWQ instrument [22] taking as reference the instruments TWQ [13], TCI [2], and GDQ [34]. The TACT instrument [9] takes as reference the TCI instru-

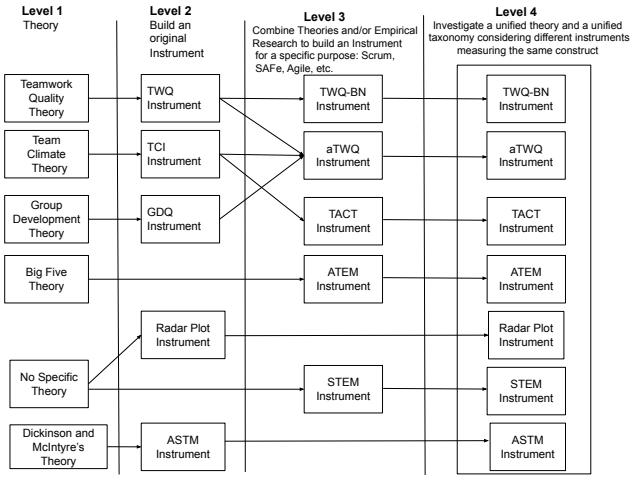


Figure 1: Evolution of Teamwork Instruments in ASD

ment [2]. Figure 1 illustrates the evolution of ASD Teamwork Instruments in ASD. In *Level 4*, we propose to investigate a unified theory and a unified taxonomy considering that we found seven ASD instruments measuring the same: the teamwork quality. Figure 1 depicts the Evolution of Agile Teamwork Instruments in ASD.

3. Study Configuration

This study presents a comprehensive study that aims to examine, compare, and synthesize eight specific instruments utilized for measuring Teamwork Quality (TWQ) in Agile Software Development (ASD). In what follows, we provide some details regarding research questions (Section 3.1). Then, we point out the research activities configuration (Section 3.2).

3.1. Research questions

We aimed to perform a quantitative and qualitative comparison between Teamwork Quality instruments in ASD, identifying trends in this comparison by focusing on the following research questions (RQs):

- **RQ1.** What is the quantitative relationship between Agile Teamwork instruments (factors and questions) and literature-based Agile Teamwork factors (themes)?
  - **RQ1.1** What are the factors used in teamwork instruments in ASD?
  - **RQ1.2** What are the most frequent factor names used in Teamwork instruments in ASD?
  - **RQ1.3** How are the frequencies of the instruments related to the work of Freire et al. [7]?
- **RQ2.** How can the Agile Teamwork instruments (factors and questions) be semantically compared in ASD?
  - **RQ2.1** What are the semantic relationships between the teamwork instrument factors in ASD?
  - **RQ2.2** What are the relationships between the evolution of teamwork instruments in ASD and the evolution of teamwork instrument factors' names and questions?

RQ1: The quantitative relationship between Agile Teamwork instruments and literature-based Agile Teamwork factors is a crucial aspect to explore in this study. Understanding how the factors and questions used in specific teamwork instruments align with established literature-based themes can shed light on the reliability and validity of these instruments. By answering RQ1.1, we can identify the factors commonly employed in teamwork instruments in Agile Software Development (ASD). This knowledge is essential as it provides a foundation for subsequent comparisons and allows researchers to focus on key aspects of teamwork assessment. RQ1.2 aims to pinpoint the most frequently utilized factor names in ASD instruments, which is valuable for understanding the prevalent themes and language employed by researchers in this field. Additionally, RQ1.3 investigates how the frequencies of instrument usage relate to the work of Freire et al. [7], a literature-based study. This comparison serves as an important validation step, enhancing the trustworthiness of the instruments' application in real-world contexts.

RQ2: The semantic comparison of Agile Teamwork instruments in ASD constitutes a fundamental aspect of this research. Semantic alignment between instrument factors and questions provides insights into the conceptual coherence and consistency of the instruments. RQ2.1 delves into the semantic relationships among teamwork instrument factors, revealing whether different instruments share common themes and concepts. This information helps researchers and practitioners in selecting the most appropriate instruments for specific assessment needs. Furthermore, RQ2.2 explores the connection between the evolution of teamwork instruments in ASD and the evolution of teamwork instrument factors' names and questions. This investigation offers valuable insights into how the instruments have evolved over time, potentially reflecting the changing nature of teamwork in the agile context. Understanding these relationships can inform future instrument development and enhance their relevance and effectiveness. By addressing RQ2, the study contributes to a deeper understanding of the nuances and intricacies of teamwork assessment, enabling researchers and practitioners to make informed decisions in their Agile Software Development projects.

3.2. Research Design

This study adopts a mixed-methods research design that combines both quantitative and qualitative approaches to achieve the research goals effectively. This dual approach enables a comprehensive exploration of the semantic relationships between specific TWQ instruments for ASD. By integrating quantitative and qualitative analyses, the study aims to provide a richer and more nuanced understanding of how these instruments are related and aligned within the agile context.

3.2.1. Data Collection

To ensure a comprehensive analysis, eight specific teamwork instruments tailored for ASD are selected (Table 1). The instruments are chosen based on their relevance and suitability to the agile context, considering their past usage and availability in the existing research literature. This rigorous selection process ensures that the chosen instruments represent the range of teamwork assessment tools applicable to ASD.

The next step involves data extraction from each selected teamwork instrument. Relevant data pertaining to the factors and questions used within each instrument is systematically gathered and organized for subsequent analysis. Additionally, literature-based Thematic Network themes identified by Freire et al. [7] are compiled to serve as a basis for comparison in the study. This addition enhances the study's depth by comparing instrument factors with established thematic themes.

3.2.2. Quantitative Analysis

In the quantitative phase, the factors present in each of the eight ASD instruments are mapped and systematically compared. This step aims to identify and highlight the most frequently utilized factors across the selected instruments. The quantitative analysis provides insights into the prevalence and significance of specific factors within the agile context.

The quantitative assessment delves further into examining instrument questions associated with each identified factor. Through frequency analysis, the study determines the prominence and prevalence of individual instrument questions for each factor. This in-depth examination helps ascertain the relative importance and weightage of different questions within the instruments.

Building on the literature-based Thematic Network themes identified by Freire et al. [7], the study compares with the factors extracted from the selected teamwork instruments. By aligning the identified factors with established thematic themes, the study seeks to identify potential overlaps, similarities, and divergences, providing a holistic view of the thematic representation within the instruments.

3.2.3. Qualitative Analysis

The qualitative phase adopts a semantic perspective to delve deeper into the context-specific characteristics of the analyzed teamwork instruments. This approach enables the identification of nuanced relationships and alignment among the instruments, considering their specificity within the agile context. The analysis also considers the theoretical underpinnings that support these instruments offering valuable insights into their semantic coherence and theoretical basis.

The qualitative analysis goes beyond descriptive exploration to identify emerging trends and patterns within the data. Drawing from the knowledge derived from the identified segments of agile instruments, the study investigates qualitative concepts that help uncover underlying themes and tendencies. This in-depth analysis supports the understanding of the semantic connections among the instruments and the contextual significance of specific factors.

3.2.4. Data Interpretation

The final phase of the study involves the interpretation of results obtained from both the quantitative and qualitative analyses. By collectively integrating the findings, the study gains a comprehensive understanding of the semantic relationships between the teamwork instruments in the context of Agile Software Development. This data interpretation stage provides valuable insights for researchers and practitioners, facilitating a coherent presentation of the findings in a manner that enhances their usability and applicability.

4. ASD instruments' factors (RQ1)

This section presents the factors we identified by analyzing the ASD instruments. It describes the computed frequencies of similar factors and discusses the results of a comparative analysis between the frequencies of ASD instrument factors and the teamwork thematic themes from the work of Freire et al. [7].

4.1. Factors of each ASD Instrument (RQ1.1)

This section provides a comprehensive description of the instrument factors identified in this study. Table 2 displays the relevant information, including the instrument name in the first column, the corresponding factor name in the second column, the symbol "#" denoting the number of questions associated with each factor in the third column, and the "Tot." representing the total number of questions for each instrument in the fourth column.

The **TWQ** instrument has six factors: Communication, Coordination, Balance of Member Contributions, Mutual Support, Effort, and Cohesion. The **Radar Plot** instrument has five factors: Shared Leadership, Team Orientation, Redundancy, Learning, and Autonomy. The **ASTM** instrument has seven factors: Team orientation, Team leadership, Monitoring, Feedback, Backup, Coordination, and Communication. The **TWQ-BN** has 17 factors: Teamwork, Team Autonomy, Cohesion, Collaboration, Self-Organizing, Coordination, Team Orientation, Communication, Daily Meetings, Team Distribution, Means of Communication, Monitoring, All Members Present, Personal Attributes, Expertise, Shared Leadership, and Team Learning. The **aTWQ** instrument has five factors: Participative safety, Support for Innovation, Vision, Task orientation, and Coordination. The **TACT** instrument has six factors: Communication, Collaboration, Leadership, Autonomy, Decision Making, and Client Involvement. The **ATEM** instrument has eight factors: Shared Mental Models, Mutual trust, Communication, Shared leadership, Peer feedback, Redundancy, Adaptability, and Team orientation. The **STEM** instrument has five factors and fourteen subfactors: Responsiveness (Refinement, Release Frequency), Stakeholder Concern (Stakeholder Collaboration, Shared Goals, Sprint Review Quality, Value Focus), Continuous Improvement (Shared Learning, Learning Environment, Psychologi-

Table 2
Teamwork Instrument Factors.

Instrum.	Factor	#	Tot.
TWQ	Communication	10	34
	Coordination	4	
	Bal.of Member Contribut.	3	
	Mutual Support	3	
	Effort	4	
	Cohesion	10	
Radar Plot	Shared Leadership	4	19
	Team Orientation	4	
	Redundancy	5	
	Learning	3	
	Autonomy	3	
ASTM	Team Orientation	2	14
	Team Leadership	2	
	Monitoring	2	
	Feedback	2	
	Backup	2	
	Coordination	2	
	Communication	2	
TWQ-BN	17 factors	17	17
aTWQ	Participative Safety	7	21
	Support for Innovation	5	
	Vision	4	
	Task Orientation	4	
	Coordination	1	
TACT	Communication	9	49
	Collaboration	7	
	Leadership	9	
	Autonomy	9	
	Decision Making	8	
	Client Involvement	7	
ATEM	Shared Mental Models	6	31
	Mutual Trust	3	
	Communication	3	
	Shared Leadership	8	
	Peer Feedback	2	
	Redundancy	3	
	Adaptability	3	
	Team Orientation	3	
STEM	Responsiveness	5	37
	Stakeholder Concern	10	
	Continuous Improvement	15	
	Team Autonomy	5	
	Management Support	2	

cal Safety, Quality, Sprint Retrospective Quality), Team Autonomy (Cross-Functionality, Self-Management), and Management Support (Management Support).

4.2. Frequency of similar factors in teamwork instruments (RQ1.2)

This section presents and analyzes the frequency of matches among the teamwork instruments. In the first step, we cross-referenced factors with identical names. For instance, Table 2 shows that both the Radar Plot instrument and the ASTM instrument have a factor named "Team Orientation." We calculated the frequency of matches for all instrument factors and presented this information in Table 3. In Column #F1 of Table 3, we listed the number of factors with the exact same name in each instrument. For example, in the "Team Autonomy" factor, a value of 1 in the TWQ-BN instrument indicates that it also has a factor named "Team Autonomy." On the other hand, Column #F2 represents cases where the factor names do not match exactly but convey the same meaning. For instance, the TACT instrument does not have an exact match for the "Team Autonomy" factor, but it does have a similar concept named "Autonomy." We accounted for this match in Column #F2. Finally, we combined the values from Column #F1 and Column #F2 into a Total column to determine that there were a total of 4 matches for the "Team Autonomy" factor.

As seen in Table 3, the "Communication" factor has 5 matches; the "Coordination", "Team Orientation", "Team Autonomy", and "Learning" factors have 4 matches; the "Collaboration", "Shared Leadership", and "Mutual Support" factors have 3 matches; the "Leadership" and "Redundancy" factors have 2 matches, and "Stakeholder Concern", "Continuous Improvement", "Team Autonomy", "Feedback", "Peer Feedback", and "Responsiveness" factors have only one match.

The factor that ranks highest with the most matches is "Communication." It secures the top position in the ranking and is present in five instruments: TWQ, ASTM, TWQ-BN, TACT, and ATEM, all of which include "Communication" within their variables. Following closely in the ranking is the "Coordination" factor with four matches. The instruments TWQ, ASTM, TWQ-BN, and aTWQ all feature a specific factor named "Coordination." As for the "Collaboration" factor, TWQ-BN, TACT, and STEM show varying degrees of matching. Specifically, TWQ-BN has one exact match, while TACT and STEM also exhibit matches. When it comes to the factors of "Shared Leadership", "Redundancy", "Feedback", and "Stakeholder Concern", STEM and ATEM present more matches than the other instruments.

When considering the frequency of occurrence among the instruments, it is worth noting that the STEM instrument stands out with the highest number of specific factors, including Team Autonomy, Continuous Improvement, Stakeholder Concern, and Responsiveness. On the other hand, the ATEM instrument features a unique factor, Peer Feedback. A comprehensive overview in Table 3 highlights the fact that only the ATEM and STEM instruments possess such distinctive factors. This observation suggests a trend towards employing more concrete factors aligned with agile-specific terminology. In contrast, the TWQ, ASTM, and TACT instruments exhibit a higher frequency of general factors, reflecting a prevalence of more generalized aspects.

We found instruments with different factor names but with the same meaning. In "Team Autonomy" factor, there is a "Autonomy" factor in the TACT instrument, a "Team Autonomy Cross-Functionality" and "Team Autonomy Self-Management" in the STEM instrument. All these questions are related to the "Team autonomy" concept but with different granularity. The STEM instrument has subfactors: Cross-Functionality and Self-Management for the "Team Autonomy" factor. In the "Learning" factor, the Radar-Plot in-

strument has a "Learning" factor. The TWQ-BN instrument has a "Team Learning" factor. The STEM instrument has the factors: "Continuous Improvement-Shared Learning" and "Continuous Improvement- Learning Environment". Note that in STEM instrument has specialized subfactors: "Shared Learning" and "Learning Environment", both related to "Continuous Improvement". In the "Collaboration" factor, the TWQ-BN and TACT instruments have the "Collaboration" factor. In STEM instruments, there is a "Stakeholder Concern-Stakeholder Collaboration" directly associated with Stakeholder Collaboration. In TWQ-BN and TACT instruments, "Collaboration" is associated with team collaboration. In the "Mutual Support" factor, The TWQ and ATEM instruments have the "Mutual Support" factor, both associated with team collaboration. In the STEM instrument, there is a factor named "Management Support" associated with the support from people in management positions.

4.3. Comparing ASD Instruments Frequencies with Freire et al. [7] teamwork thematic network (RQ1.3)

Freire et al. [7] presented a literature-based Thematic Network identifying themes and codes shown in Table 4. For example, the theme "Coordination" is related to the codes "Coordination", "Performance Monitoring", "Task Novelty" and "Familiarity". Table 4 shows that the most frequent theme in the agile teamwork literature is "Team Orientation" with 22 matches, followed by "Coordination" with 16 matches. The third most frequent is "Expertise" with 15 matches, and so on. By comparing the results shown in Table 4 and Table 3, we identified an important trend: "Team Orientation" and "Coordination" are in the top 3 ranking in both, Freire et al.'s work and our work, indicating that they are key factors for agile teamwork quality.

5. Semantic Comparison between ASD Instruments (RQ2)

This section discusses the results of the semantic comparison between the instruments (Section 5.1) and investigates the relationship between the evolution of teamwork instruments in ASD and the evolution of teamwork instruments factors names and questions (Section 5.2).

5.1. Semantic relationship between teamwork instruments factors in ASD (RQ2.1)

As previously discussed, Table 3 shows the frequency in which we identified each of Freire et al.'s factors in the agile teamwork instruments under study. For example, we identified the factor "Communication" in five instruments: TWQ, ASTM, TWQ-BN, TACT, and ATEM. This result indicates that such instruments are similar in terms of containing questions related to such a factor. To address RQ2.1, we went beyond and performed qualitative analysis on the questions of each instrument that were mapped to such a factor to consider semantic aspects and have a more in-depth comparison between the instruments under study.

Table 3
Frequency in each Instruments Factors.

Instrum. Factor	Instrum.	#F1	#F2	Tot.
Communication	TWQ	1	0	5
	ASTM	1	0	
	TWQ-BN	1	0	
	TACT	1	0	
	ATEM	1	0	
Coordination	TWQ	1	0	4
	ASTM	1	0	
	TWQ-BN	1	0	
	aTWQ	1	0	
	ATEM	0	0	
Team Orientation	Radar Plot	1	0	4
	ASTM	1	0	
	TWQ-BN	1	0	
	ATEM	1	0	
Team Autonomy	TWQ-BN	1	0	4
	TACT	0	1	
	STEM	2	0	
Learning	Radar Plot	1	0	4
	TWQ-BN	0	1	
	STEM	0	2	
Collaboration	TWQ-BN	1	0	3
	TACT	1	0	
	STEM	0	1	
Shared Leadership	Radar-Plot	1	0	3
	TWQ-BN	1	0	
	ATEM	1	0	
Mutual Support	TWQ	1	0	3
	ATEM	1	0	
	STEM	0	1	
Leadership	TACT	1	0	2
	ASTM	1	0	
Redundancy	Radar Plot	1	0	2
	ATEM	1	0	
Stakeholder Concern	STEM	1	0	1
Continuous Improvement	STEM	1	0	1
Feedback	ASTM	1	0	1
Peer Feedback	ATEM	1	0	1
Responsiveness	STEM	1	0	1

5.1.1. Communication

For the "Communication" factor, we compared TWQ, ASTM, TWQ-BN, TACT, and ATEM. Ten questions from Instrument 1 (TWQ) focus on team communication (Questions 1, 2, 3, 4, 5, 6, 7, 8, 9, 10), while Question 13 from Instrument 3 (ASTM) is related to verifying information before making a report. Effective communication and information exchange are also implicit in the questions from Instrument 6 (TACT) and Instrument 4 (TWQ-BN), as they inquire about freely talking, updating lists, listening to opinions, and knowing team members' skills. By analyzing these questions, we identified the following themes:

Openness and Transparency: Three Questions from Instrument 1 (TWQ) address the openness of communication (Questions 5, 6, 7), and Instrument 6 (TACT) emphasizes

Table 4
ASD Theme frequencies in Freire et al. work

ASD Theme	ASD Code	#	Tot.
Team Orientation	Orientation	7	22
	Value Diversity	1	
	Goals	2	
	Roles	2	
	Holistic Team Involvement	1	
	Experience in the Organi.	1	
	Trust	5	
	Motivation	1	
	Norms	2	
Coordination	Coordination	5	16
	Performance Monitoring	9	
	Task Novelty	1	
	Familiarity	1	
Expertise	Tools knowledge	2	15
	Collective Knowledge	4	
	Adequate Skills	1	
	Redundancy	7	
	Experience with Work	1	
Management Mechanisms	Management	4	10
	Planning	1	
	Discussion	1	
	Implementation	1	
	Evaluation	1	
	Information Radiators	1	
	Decision Making	1	
Shared Leadership	Shared Leadership	8	9
	Formal Leadership	1	
Communication	Communication	9	9
Organization Culture	Culture	4	8
	Structure	1	
	Team Size	2	
	Organization Support	1	
Collaboration	Interdependence	1	8
	Collaboration	7	
Learning	Learning	8	8
Members Personality	Individual Differences	1	5
	Heterogeneity	1	
	Personality	3	
Team Autonomy	Autonomy	4	5
	Task Control	1	
Feedback	Awareness	1	5
	Acceptance	1	
	Feedback	3	
Cohesion	Cohesion	3	3

openness in freely talking about difficulties (Question 1).

Team Interaction and Understanding: Question 6 from Instrument 1 (TWQ) and Question 6 from Instrument 6 (TACT) both relate to understanding team members' skills and expertise and using them appropriately.

Project Progress and Information: Questions from Instrument 7 (ATEM) focus on project progress and information visualization (Questions 10, 11, 12), while Instrument 6 (TACT) has a question related to knowing project problems and team difficulties through daily meetings (Question 7).

Information Accuracy and Precision: Question 9 from Instrument 1 (TWQ) and Question 9 from Instrument 6 (TACT) inquire about the precision and scope of information received.

5.1.2. Coordination

For the "Coordination" factor, we compared questions from the following instruments: TWQ, ASTM, TWQ-BN, aTWQ, and ATEM. Next, we present our results grouped by the main themes identified while analyzing the questions.

Task Coordination: Questions from Instrument 1 (TWQ) and Instrument 5 (aTWQ) both focus on task coordination and harmonization: Question 11 from TWQ: "The work done on subtasks within the project was closely harmonized". Question 21 from aTWQ: "Is there a common understanding when working on parallel subtasks and agreement on common work breakdown structures, schedules, budgets, and deliverables?".

Clarity and Acceptance of Goals: Instrument 1 (TWQ) and Instrument 3 (ASTM) include questions related to goal clarity and acceptance: Question 12 from TWQ: "There were clear and fully comprehended goals for subtasks within our team". Question 11 from ASTM: "Passing performance-relevant data to other members efficiently". Question 12 from ASTM: "Facilitating the performance of other members' jobs".

Synchronization and Integration of Tasks: Instrument 4 (TWQ-BN) has a question that relates to the synchronous and integrated execution of tasks: Question 6 from TWQ-BN: "The team executes its tasks in a synchronous and integrated manner".

Conflict and Diverging Interests: Instrument 1 (TWQ) includes a question about conflicting interests regarding subtasks/subgoals: Question 14 from TWQ: "There were conflicting interests in our team regarding subtasks/subgoals". These are some of the semantic similarities between the questions from the different instruments. The themes of task coordination, goal clarity and acceptance, task synchronization, and conflict are present in the questions.

5.1.3. Team Orientation

For the "Team Orientation" factor, we compared RadarPlot, ASTM, TWQ-BN, and ATEM. By analyzing these instruments questions, we identified the following themes:

Valuing and Considering Alternative Suggestions: Questions from Instrument 2 (Radar Plot - Team Orientation) and Instrument 7 (ATEM-TC-Team Orientation) focus on how the team values and considers alternative suggestions: Question 5 from Radar Plot: How does the team take into account alternative suggestions in team discussions? Question 6 from Radar Plot: How does the team value alternative suggestions? Question 29 from ATEM: "Taking into account alternative solutions provided by teammates and appraising that input to determine what is most correct".

Participation and Commitment to Team Goals: Instrument 3 (ASTM - Team Orientation) and Instrument 7 (ATEM-TC-Team Orientation) include questions related to team member participation and commitment to team goals: Question 1 from ASTM: assigning a high priority to team goals. Question 2 from ASTM: Participate willingly in all relevant as-

pects of the team. Question 30 from ATEM-TC: Increased task involvement, information sharing, strategizing, and participatory goal setting.

Trust and Collaboration: Question 7 from Instrument 4 (TWQ-BN - Team Orientation) and Question 31 from Instrument 7 (ATEM-TC-Team Orientation) touch on trust and collaboration within the team. Question 7 from TWQ-BN: The team members trust each other and feel motivated to work together to achieve the team's goals. Question 31 from ATEM-TC: The team sticks together and remains united.

Task and Individual Relations: Instrument 2 (Radar Plot - Team Orientation) includes questions that inquire about the relationship between team members and their tasks. Question 7 from Radar Plot: How do team members relate to the tasks of individuals? Question 8 from Radar Plot: What kind of ownership do the team members have to the project? These are some of the semantic similarities between the questions from the different instruments. The themes of valuing alternative suggestions, participation in team goals, trust, collaboration, and task relations are present in the questions.

5.1.4. Team Autonomy

For the "Team Autonomy" factor, we compared TWQ-BN, TACT, and STEM and identified the following themes:

Autonomy in Decision Making and Planning: Questions from Instrument 6 (TACT - Autonomy) and Instrument 8 (STEM - Team Autonomy) both focus on autonomy in decision-making and planning: Question 28 from TACT: In the current project, I can choose the tasks I want to execute in the iteration. Question 34 from TACT: My team has the decision authority and responsibility to plan the iteration. Question 36 from STEM: Most people in this team have the ability to solve the problems that come up in their work. Question 38 from STEM: This team has control over the scheduling of teamwork.

Autonomy in Technical Solutions: Instrument 6 (TACT - Autonomy) and Instrument 4 (TWQ-BN - Team Autonomy) have questions related to autonomy in technical solutions: Question 30 from TACT: In this organization, we can suggest changing the team's software process development. Question 33 from TACT: My team can communicate with the product owner and other relevant stakeholders. Question 2 from TWQ-BN: No external agent is interfering with how the team executes its tasks. The external agent collaborates with them to define what will be.

Protection of Team Autonomy: Instrument 6 (TACT - Autonomy) includes a question about the team facilitator protecting the team's autonomy from external interferences: Question 29 from TACT: "In the current project, the team facilitator protects the team autonomy from external interferences". These are some of the semantic similarities between the questions from the different instruments. The themes of autonomy in decision-making, planning, technical solutions, communication, and protection of team autonomy are present in the questions but keep in mind that this analysis is based on the questions provided, and there may be other connections and interpretations depending on the specific usage and context of these instruments.

5.1.5. Learning

For the "Learning" factor, we compared Radar-Plot, TWQ-BN, and STEM and identified the following themes:

Learning and Improvement in Software Development: Questions from Instrument 2 (Radar Plot - Learning) directly relate to learning and improvement in software development: [14] from Radar Plot: What are the arenas where you give feedback on each other's work? [15] from Radar Plot: "How are software development problems identified, and do you improve the development method?" [16] from Radar Plot: Do you keep what works well in your development process? [17] from Radar Plot: "How are artifacts in the development process (burndown chart, backlog, daily meetings, sprint reviews, and retrospectives) used to learn?"

Team Learning and Adaptation: Instrument 4 (TWQ-BN - Team Learning) has a question related to team learning and adaptation: [17] from TWQ-BN: The team adapts itself to changes in the team environment and adjusts the strategies as needed.

Shared Learning and Collaboration: Instrument 8 (STEM - Continuous Improvement - Shared Learning) includes questions related to shared learning and collaboration with other teams: [21] from STEM (Continuous Improvement - Shared Learning): This team frequently works with other groups or teams to solve shared problems; [22] from STEM (Continuous Improvement - Shared Learning): Teams in this organization share what they learn with other teams; [23] from STEM (Continuous Improvement - Shared Learning): Members of this team frequently meet with other teams to identify improvements.

Learning Environment and Support for Learning: Instrument 8 (STEM - Continuous Improvement - Learning Environment) also has questions related to the learning environment and support for learning: [24] from STEM (Continuous Improvement - Learning Environment): In and around this team, people are given time to support learning; [25] from STEM (Continuous Improvement - Learning Environment): In and around this team, people are rewarded for learning.

The Radar Plot questions focus on aspects of software development processes and feedback mechanisms, while the STEM questions explore how teams collaborate, share knowledge, and support learning. The TWQ-BN question touches on the team's adaptability and strategy adjustments in response to changes in the team environment.

5.1.6. Collaboration

For the "Collaboration" factor, we compared TWQ-BN, TACT, and STEM. Question 4 (TWQ) and Questions 10, 11, 12, 13, 14, 15, and 16 (TACT) all revolve around teamwork, collaboration, and how team members work together to achieve common goals. As a result of analyzing such questions, we identified the following themes:

Project Development: Question 4 (TWQ) talks about success on project development, and some questions from

TACT (e.g., Questions 14, 15, 16) mention specific aspects related to projects, such as project-related decisions, problem analysis, and software design.

Team Support: Questions 4 (TWQ) and Questions 11, 12, and 13 (TACT) highlight the aspect of team members helping each other and providing support whenever needed.

Knowledge Sharing: Question 10 (TACT) indicates team members' consideration of sharing know-how with each other, which might be related to the collaboration and success mentioned in Question 4 (TWQ).

Semantic Similarities for Collaboration: The questions from TACT (10, 11, 12, 13, 14, 15, and 16) are all related to different aspects of team collaboration. They cover topics like knowledge sharing, mutual support, efficient teamwork, consistent decision-making, problem analysis, and software design based on user stories. These factors indicate a strong emphasis on collaboration and teamwork within the team.

Semantic Similarities for Stakeholder Concern - Stakeholder Collaboration: The questions from STEM (11, 12, and 13) all revolve around the team's interactions with stakeholders, users, and customers. They suggest a high level of engagement and collaboration between the team and external parties. These factors indicate that the team is attentive to stakeholder needs and actively seeks their input and collaboration.

Overall, the semantic similarities between the questions can be summarized as follows: TWQ-BN and TACT instruments focus on collaboration within the team. TWQ-BN specifically mentions "a high degree of collaboration", while TACT addresses various collaboration aspects like knowledge sharing, support, efficient teamwork, and decision-making. The STEM instrument, on the other hand, emphasizes stakeholder concern and collaboration. It highlights the team's interactions with stakeholders, customers, and users, indicating a strong focus on understanding and meeting their needs. In conclusion, the instruments TWQ-BN, TACT, and STEM all share the theme of collaboration, but they approach it from different angles. TWQ-BN emphasizes collaboration within the team, while TACT covers various aspects of team collaboration. STEM, on the other hand, emphasizes stakeholder concern and the team's collaboration with external parties.

5.1.7. Shared Leadership

For the "Shared Leadership" factor, we compared Radar-Plot, TWQ-BN, and ATEM. After analyzing these instruments' questions, we identified the following themes:

Decision-Making and Empowerment: Questions from Instrument 1 (Radar Plot - Shared Leadership) and Instrument 2 (ASTM - Team Leadership) focus on decision-making and empowerment within the team: Question 1 from Radar Plot: Is everyone involved in the decision-making process? Question 2 from Radar Plot: "Do team members make important decisions without consulting other team members?" Question 3 from ASTM: explaining to other team members exactly what is needed from them during an assignment. Question 4 from ASTM: listening to the concerns of other team members. Shared Decision Authority and Leadership: Instrument 3 (TWQ-BN - Shared Leadership) has a question related to shared decision authority and leadership: Question 16 from TWQ-BN: The decision authority and leadership are shared.

Team Facilitation: Instrument 4 (TACT - Leadership) focuses on team facilitation and the role of a team facilitator: Questions 17 to 25 from TACT include various aspects of team facilitation, such as providing helpful feedback, eliminating barriers, listening to team ideas and concerns, discussing team problems, protecting the team from outside interference, helping the team acknowledge and solve disagreements, and assisting in understanding iteration objectives.

Agile Team Practices: Instrument 5 (ATEM-TC-Shared Leadership) is centered around agile team practices and servant leadership: Questions 13 to 20 from ATEM-TC focus on various aspects of agile team practices, such as team problem-solving, determining performance expectations, and interaction patterns, synchronizing and combining individual contributions using agile practices and automated tools, seeking and evaluating information affecting team functioning, determining team member roles based on agile values and methodologies, determining the frequency and type of preparatory meetings and feedback sessions, and the role of a servant leader in facilitating a boundary-spanning function. These are some of the semantic similarities between the questions from the different instruments. The themes of decision-making, empowerment, shared leadership, team facilitation, and agile practices are present in the questions.

5.1.8. Mutual Support

For the "Mutual Support" factor, we compared TWQ, ATEM, and STEM. We identified the following questions related to this factor: [18] TWQ: "The team members helped and supported each other as best they could."; [19] TWQ: "If conflicts came up, they were easily and quickly resolved."; [20] TWQ: "Discussions and controversies were conducted constructively."; [7] ATEM-TCM: "Mutual trust - Information sharing."; [8] ATEM-TCM: "Mutual trust - Willingness to admit mistakes and accept feedback."; [9] ATEM-TCM: "Mutual trust - Supportive team social climate."

Mutual support and Trust: Questions [18], [19], and [20] from TWQ and Questions [7], [8], and [9] from ATEM, all address different aspects of mutual support and trust within the team. TWQ focuses on supporting each other, resolving conflicts, and constructive discussions, while ATEM highlights mutual trust through information sharing, feedback acceptance, and a supportive social climate.

Management support: STEM contains questions related to management support: [41] STEM: "People in a management position generally understand why this team works with Scrum."; [42] STEM: "People in a management position help this team work with Scrum.". Questions [41] and [42] from STEM, both pertain to management support in the context of the team working with Scrum. They suggest that people in a management position know the team's utilization of Scrum

and provide assistance in this regard.

Overall, TWQ (Instrument 1) and ATEM (Instrument 7) emphasize aspects of mutual support and trust within the team. While TWQ addresses support, conflict resolution, and constructive discussions, ATEM focuses on information sharing, feedback acceptance, and a supportive team social climate. STEM (Instrument 8) questions center around management support, particularly regarding the team's use of Scrum.

5.1.9. Leadership

In the "Leadership" factor, we compared TACT and ASTM. Based on the questions provided by Instrument 3 (ASTM) and Instrument 6 (TACT), we identified s emantic similarities in the next questions. Team Leadership: [3] ASTM: "Explaining to other team members exactly what is needed from them during an assignment."[4] ASTM: "Listening to the concerns of other team members." [17] TACT: "In the current project, the team, the product owner, and the team facilitator work excellently together to plan the iteration." [18] TACT: "The team facilitator gives me helpful feedback on how to be more effective." [19] TACT: "The team facilitator eliminates barriers, encourages, and facilitates the use of agile methods." [20] TACT: "The team facilitator listens to my ideas and concerns." [21] TACT: "The team facilitator discusses the problems of the team." [22] TACT: "The team facilitator protects the team from outside interference." [23] TACT: "The team facilitator helps my team to acknowledge and solve our disagreements." [24] TACT: "The team facilitator assists in understanding whether the iteration objectives are clear and whether the team agrees with these objectives." [25] TACT: "The team facilitator gives the team helpful feedback on how to be more agile."

Both ASTM (3 and 4 questions) and TACT (17 to 25 question) instruments include questions related to team leadership. ASTM focuses on team leadership involving explaining assignments clearly and listening to team members' concerns. TACT addresses leadership in the context of the team facilitator's role and their collaboration with the team and product owner. The TACT questions highlight various aspects of effective l eadership, s uch a s p roviding feedback, encouraging agile methods, protecting the team, resolving disagreements, and promoting agility.

Overall, ASTM (Instrument 3) and TACT (Instrument 6) have questions related to team leadership. ASTM focuses on leadership involving task explanation and listening to concerns, while TACT addresses leadership in the context of the team facilitator's role and their impact on the team's performance, collaboration, and agile practices.

In conclusion, the instruments ASTM and TACT touch on different a spects o f t eam l eadership. A STM addresses leadership in terms of task communication and listening, while TACT emphasizes the team facilitator's role and their influence on team dynamics, problem-solving, and agile practices.

5.1.10. Redundancy

For the "Redundancy" factor, we compared Radar Plot and ATEM. Based on the questions provided by Instrument 2 (Radar-Plot) and Instrument 7 (ATEM), we seek to identify the semantic similarities between them: Redundancy: [9] Radar-Plot: "How easy is it to complete someone else's task?"; [10] Radar-Plot: "If you are stuck, do you get help?"; [11] Radar-Plot: "Do you help others when they have problems?"; [12] Radar-Plot: "How are tasks allocated?"; [13] Radar-Plot: "If someone leaves the team, is it easy to substitute this person?"; [23] ATEM-TC: "Recognition by potential backup providers that there is a workload distribution problem in their team."; [24] ATEM-TC: "Shifting of work responsibilities to underutilized team members."; [25] ATEM-TC: "Completion of the whole task or parts of tasks by other team members." Semantic Similarities for Redundancy: The questions from both Radar-Plot (9 to 13) and ATEM-TC (23 to 25) instruments touch on the concept of redundancy within the team. Radar-Plot questions focus on how easy it is to complete each other's tasks, provide help, and allocate tasks. They also inquire about the ease of substituting team members if needed. On the other hand, ATEM questions address redundancy in terms of recognizing workload distribution issues, shifting work responsibilities, and task completion by other team members.

Overall, both Radar-Plot (Instrument 2) and ATEM (Instrument 7) have questions related to redundancy within the team. Radar-Plot addresses the ease of completing tasks, providing help, task allocation, and substitution of team members. ATEM questions highlight redundancy in terms of recognizing workload issues, shifting responsibilities, and task completion by other team members. In conclusion, the instruments Radar-Plot and ATEM touch on different aspects of redundancy within the team. Radar-Plot addresses the ease of task completion and support, while ATEM emphasizes workload distribution, task shifting, and task completion by various team members.

5.1.11. Stakeholder Concern

For the "Stakeholder Concern" factor, we investigated the STEM instrument. Based on the questions provided by Instrument 8 (STEM), we identified the semantic similarities between the following questions: Stakeholder Collaboration: [11] STEM: "Members of this team frequently meet with users or customers of what this team creates."; [12] STEM: "People from this team often invite or visit people that use what this team works on."; [13] STEM: "People in this team closely collaborate with users, customers, and other stakeholders."; Shared Goals: [14] STEM: "This team generally has clear Sprint Goals."; [15] STEM: "During Sprint Planning, this team formulates a clear goal for the Sprint."; Sprint Review Quality: [16] STEM: "The Product Owner of this team uses the Sprint Review to collect feedback from stakeholders."; [17] STEM: "During Sprint Reviews, stakeholders frequently try out what this team has been working on during the Sprint." Value Focus: [18] STEM: "The Product Owner of this team has a clear vision for the product.";

[19] STEM: "The Product Backlog of this team is ordered with a strategy in mind."; [20] STEM: "Everyone in this team is familiar with the vision for the product.";

Further, STEM's questions [16] and [17] are related to the quality of Sprint Reviews. They discuss the involvement of stakeholders in providing feedback and trying out the team's work during the Sprint Review, indicating a focus on gathering valuable input from stakeholders. Value Focus: Questions [18], [19], and [20] relate to the team's value focus. They touch on aspects such as the Product Owner having a clear vision for the product, the strategic ordering of the Product Backlog, and everyone in the team being familiar with the product's vision. These questions suggest a strong orientation toward delivering value to stakeholders.

The questions from Instrument 8 (STEM) can be grouped into several categories based on their similarities: Stakeholder Collaboration: Questions [11], [12], and [13] all pertain to stakeholder collaboration. They highlight the team's frequent interactions with users, customers, and other stakeholders, focusing on engaging and working closely with them. Shared Goals: Questions [14] and [15] revolve around shared goals. They address the team's clarity on Sprint Goals and the formulation of clear goals during Sprint Planning, which indicates a strong emphasis on having well-defined objectives.

Overall, STEM (Instrument 8) questions address stakeholder collaboration, shared goals, sprint review quality, and value focus. The instrument focuses on actively involving stakeholders, defining clear goals, obtaining valuable feedback during reviews, and delivering value through a well-defined product vision and ordered b acklog. In conclusion, the instrument STEM (Instrument 8) focuses on various aspects of stakeholder engagement, goal-setting, review quality, and value-driven development, all contributing to effective project execution and successful product delivery.

5.1.12. Continuous Improvement

For the "Continuous Improvement" factor, we investigated the STEM instrument. Based on the questions provided by Instrument 8 (STEM), we identified the following themes:

Shared Learning: [20] STEM: "This team frequently works with other groups or teams to solve shared problems."; [21] STEM: "Teams in this organization share what they learn with other teams."; [22] STEM: "Members from this team frequently meet with other teams to identify improvements."; Continuous Improvement - Learning Environment: [23] STEM: "In and around this team, people are given time to support learning."; [24] STEM: "In and around this team, people are rewarded for learning.";

Psychological Safety: [25] STEM: "In and around this team, people give open and honest feedback to each other."; [26] STEM: "In and around this team, people listen to others' views before speaking."; [27] STEM: "In and around this team, whenever people state their view, they also ask what others think."; [28] STEM: "In and around this team, people openly discuss mistakes to learn from them."; [29] STEM: "In and around this team, people help each other learn.".

Quality: [30] STEM: "Members of this team have a shared understanding of what quality means to them."; [31] STEM: "People in this team frequently talk about quality and how to improve it.";

Sprint Retrospective Quality: [32] STEM: "The Sprint Retrospectives of this team generally result in at least one useful improvement."; [33] STEM: "During Sprint Retrospectives, this team openly discusses improvements." Semantic Similarities: The questions from Instrument 8 (STEM) can be grouped into several categories based on their similarities.

Shared Learning: Questions [20], [21], and [22] all focus on shared learning and collaboration. They highlight how the team works with other groups or teams, shares knowledge within the organization, and engages in cross-team meetings to identify improvements. Continuous Improvement - Learning Environment: Questions [23] and [24] pertain to the learning environment. They address the provision of time and rewards for supporting learning, which fosters a culture of continuous improvement.

Psychological Safety: Questions [25] to [29] all relate to psychological safety. They emphasize the importance of open and honest feedback, active listening, inviting others' views, openly discussing mistakes, and helping each other learn. Continuous Improvement - Quality: Questions [30] and [31] are related to the team's understanding of quality and how they frequently discuss it and work to improve it.

Sprint Retrospective Quality: Questions [32] and [33] focus on the quality of Sprint Retrospectives. They mention the usefulness of improvements resulting from these retrospectives and the team's open discussions during them.

Overall, STEM (Instrument 8) questions address various aspects of continuous improvement. They cover shared learning and collaboration with other teams, creating a supportive learning environment, fostering psychological safety for open communication, discussing quality improvements, and the effectiveness of Sprint Retrospectives in generating useful insights.

In conclusion, the instrument STEM (Instrument 8) highlights different dimensions of continuous improvement within the team, encompassing shared learning, learning environment, psychological safety, quality discussions, and Sprint Retrospective effectiveness. These factors collectively contribute to the team's ability to continuously learn, evolve, and deliver value.

5.1.13. Feedback and Peer Feedback

For the "Feedback" factor, we investigated the ASTM instrument. For the "Peer Feedback" factor, we investigated the ATEM instrument. Based on the questions provided by Instrument 3 (ASTM) and Instrument 8 (STEM), we identified the following questions: [7] ASTM: "Responding to other members' requests for information about their performance."; [8] ASTM: "Accepting time-saving suggestions offered by other team members."; [21] ATEM-TC: "Identifying mistakes and lapses in other team members' actions.";

[22] ATEM-TC: "Regular feedback regarding team member actions to facilitate self-correction."

The questions from ASTM (7 and 8) and ATEM-TC (21 and 22) instruments focus on different aspects of feedback within the team: ASTM questions emphasize the exchange of feedback between team members. Question 7 addresses how team members respond to requests for performance-related information, while Question 8 focuses on their receptiveness to time-saving suggestions provided by others. ATEM-TC questions focus on peer feedback within the team. Question 21 mentions identifying mistakes and lapses in other team members' actions, indicating a form of feedback that helps in recognizing areas for improvement. Question 22 highlights the importance of regular feedback to facilitate self-correction, suggesting an ongoing feedback loop to enhance team performance.

Overall, ASTM (Instrument 3) and ATEM (Instrument 8) have questions related to feedback within the team. ASTM focuses on responding to information requests and accepting suggestions, while ATEM emphasizes the identification of mistakes, providing regular feedback, and facilitating self-correction.

In conclusion, the instruments ASTM and ATEM address different aspects of feedback within the t eam. ASTM highlights feedback exchange and acceptance of suggestions, while ATEM focuses on peer feedback for recognizing errors and supporting ongoing improvement through regular feedback.

5.1.14. Responsiveness

In the "Responsiveness" factor, we analyzed the STEM instrument (Instrument 8). We intended to identify the semantic similarities within the provided questions. In the domain of Responsiveness and Refinement, the questions were: [6] STEM: "The team's Sprint Backlog typically comprises numerous small items."; [7] STEM: "This team allocates time during the Sprint to elaborate on the work slated for the succeeding Sprints."; and [8] STEM: "Throughout the Sprint, this team commits time to decompose work for upcoming Sprints."

Regarding Responsiveness and Release Frequency, the questions were: [9] STEM: "The bulk of this team's Sprints lead to software that is prepared for deployment to production."; [10] STEM: "For this team, the majority of Sprints culminate in an increment ready for user release."

For Responsiveness and Refinement, questions [6], [7], and [8] collectively denote the team's adaptability and responsiveness. They underline the team's approach of maintaining a Sprint Backlog with numerous smaller items and dedicating time within the Sprint to clarify and decompose work for future Sprints. Responsiveness and Release Frequency are addressed in questions [9] and [10], where the focus is on the team's aptitude to regularly produce software or increments that can be released to users, showcasing the team's capacity to deliver value frequently.

In essence, the STEM instrument (Instrument 8) queries examine various facets of responsiveness. Questions related to refinement underscore the team's competency in decomposing and elucidating work during the Sprint, enabling adaptability. Questions associated with release frequency emphasize the team's consistent delivery of software or increments prepared for deployment or user release.

In summary, Instrument 8 (STEM) emphasizes distinct aspects of responsiveness, including refinement practices that foster adaptability and the team's ability to deliver valuable software or increments regularly. These factors collectively enhance the team's agility and capacity to deliver user value.

5.2. Relationship between the evolution of teamwork instruments in ASD and evolution of teamwork instruments factors names and questions (RQ2.2)

Given the results presented in Section 5, we found that the instruments ATEM, STEM, aTWQ and TWQ-BN brought new concepts directly associated with the agile context, among them: daily meetings, retrospective meetings, and Sprint Review. STEM brought other concepts like Cross-Functionality and Self-Management associated with Team Autonomy.

We suggest classifying agile teamwork instruments into two groups: Generic teamwork instruments and Agile-based teamwork instruments. The generic ones were developed until 2018: TWQ, Radar Plot, and ASTM. The Agile-based ones were developed later: TWQ-BN, aTWQ, TACT, ATEM, and STEM. We noted that the factors and questions from the Agile-based one included a terminology closely related to agile concepts. Further, ATEM (with seven factors) and STEM (with five factors and 14 subfactors) present a trend toward increasing the number of factors and subfactors compared to the older instruments.

6. Discussion and Findings

This section discusses this study's research questions and the trends observed. In summary, we mapped the factors of the eight teamwork instruments, then we compared them with the Themes found by Freire et al. [7]. The objective was to understand how the themes and instrument questions are quantitatively related. Then, we intended to identify trends in these factors. Considering the Themes analysis in Section 4.3. The results showed that Team Orientation and Coordination were identified among the top three rankings, both in the frequency of instrument questions and the frequencies of literature-based Thematic Network developed in Freire et.al [7].

We found in our semantic analysis important themes associated a many instrument factors. In Communication we found the themes: Openness and Transparency, Team Interaction and Understanding, Project Progress and Information, Information Accuracy and Precision. In Coordination factor, we found: Task Coordination, Clarity and Acceptance of Goals, Synchronization and Integration of Tasks, etc. In Team Orientation we found: Valuing and Considering Alternative Suggestions, Participation and Commitment to Team Goals, Trust and Collaboration, Task and Individ-

ual Relations, etc. In Collaboration we found: Team Support, Knowledge Sharing, etc. In Mutual Support, we found: Trust, and Management support. The present study can be a starting point for the development of new studies exploring the relationships between the instruments' factors and the themes identified in this study.

The researchers could investigate whether lower frequencies are, in fact, less important for teamwork quality. In this way, researchers will already know which subparts of the instruments to use. It was found the frequency of appearance of each factor related to the teamwork quality and the number of corresponding questions for each instrument. With this knowledge, this work can support other works that need to use ASD teamwork instruments for a specific p urpose. For example, if a researcher needs to investigate the relationship between Communication and Shared Leadership in a company, he can choose specific A SD i nstruments: F or Communication (TWQ, ASTM, TWQ-BN, TACT, and ATEM) and Shared Leadership (Radar Plot, TWQ-BN, and ATEM) in the investigation based on the requirements. Qualitative concepts can be investigated in future works focusing on investigating the ASD factors from the knowledge of the identified parts of the agile instruments.

This study can support using a Teamwork Instrument for a specific p urpose. For example, if a researcher needs to investigate the relationship between Feedback and Team Autonomy, he can choose what parts of the instruments to use. This work highlights that the ASD literature codes: Task Control, Communication, Coordination, and Team Autonomy are the most used in ASD Teamwork Instruments. This is an important result, as it confirms t hat t he factors identified by Freire et al. [7] are, in fact, those that are being used more frequently in specific A SD i nstruments, w hich were developed based on strong literature theories and empirical studies. Additionally, we identified a nd c ompared t he referred questions in the eight ASD instruments analyzed in this work. We noted that finding a standard terminology for ASD Teamwork factors remains challenging, and there is a need for further investigation into this area. Finally, practitioners can benefit from the study's findings by better understanding the importance of Teamwork instruments in ASD.

7. Limitations and threats to validity

In this study, we explored various validity threats that may arise during the realization of our research, encompassing internal, external, construct, and conclusion validity.

Regarding internal validity, potential issues may arise from selection bias, history effects, instrumentation, and maturation. To mitigate these threats, we employed random sampling techniques and defined clear inclusion criteria for selecting ASD instruments and research articles. Additionally, we carefully control external events and changes by collecting data over a consistent time period and conducting longitudinal studies. Standardization and pilot testing of instrument administration and interpretation help address potential instrumentation concerns.

Construct validity threats may arise from conceptual clarity, instrument validity, and measurement errors. We take measures to address these concerns by providing a clear definition of the constructs of interest and employing a conceptual framework. Established and validated teamwork instruments are used to ensure accurate measurement of constructs. Additionally, we employed reliable data collection methods and appropriate techniques to minimize measurement errors.

External validity threats revolve around generalizability and timeframe relevance. To address these concerns, we clearly define the target population and context of our study. Efforts are made to replicate real-world conditions in the study design to enhance validity. We ensure that data collection and analysis are up-to-date and reflect current practices in the field. Moreover, the study relies on solid theories that support the analyzed teamwork instruments. The results may not fully capture the variability or applicability of other theoretical frameworks, potentially limiting the external validity of the conclusions to different theoretical perspectives.

Regarding conclusion validity, the study analyzes eight specific teamwork instruments for Agile Software Development (ASD). The findings may not fully represent the entire population of ASD instruments, potentially limiting the generalizability of the results to other instruments that were not included in the analysis. Moreover, the study focuses on teamwork instruments specifically designed for an agile context. The results may not directly apply to teamwork instruments used in non-agile contexts, reducing the external validity of the findings for broader applications.

By proactively addressing these validity threats and implementing appropriate actions, we aim to enhance the quality and reliability of our study, providing more robust and meaningful findings for the scientific community.

8. Implications

In light of the findings from this study, we have identified several implications for both research and practice in the context of measuring TWQ in ASD.

Implications for research. This study sheds light on the evolution of TWQ instruments, providing valuable insights for further research. The findings highlight the existence of multiple models with different constructs and measures for assessing TWQ and TWE. This prompts researchers to delve deeper into understanding the relationships between these instruments and how they have evolved over time. The study also emphasizes the need for standardization of terminology, as semantically similar factors are often labeled differently across instruments. This calls for future research to focus on developing a conceptual framework that integrates instrument factors within the agile context, facilitating better alignment and comparison of results. Moreover, the identified gaps and specialized factors specific to the agile context present opportunities for researchers to develop new instruments and further advance the understanding of teamwork in ASD.

Implications for practice. The findings of this study hold practical implications for organizations engaged in ASD. Classifying teamwork instruments into generic and specific agile instruments guides practitioners in selecting appropriate instruments based on their specific context and requirements. The evolution of instruments with specialized factors underscores the importance of considering these factors when evaluating and managing teamwork in agile projects. Furthermore, the identified need for terminology standardization emphasizes the importance of consistent and clear team communication. Organizations can benefit from adopting a unified taxonomy derived from this research to ensure consistent understanding and usage of teamwork concepts. The study also emphasizes the value of developing new instruments that align with the agile context, allowing organizations to assess and improve their teamwork practices effectively. Overall, the insights gained from this study can inform and guide practitioners in selecting and implementing appropriate teamwork instruments and strategies to enhance collaboration and team performance in ASD projects.

9. Final Remarks

Our study significantly contributes to the teamwork literature by exploring the relationship between ASD literature-based codes identified by Freire et al. [7] and Agile Instruments factors in ASD. By comparing eight specific ASD instruments and showcasing the frequency of matches, we offer insights that can inform further research. Moreover, our identification of ASD instrument questions through semantic analysis enables broader coverage for future studies, potentially leading to new discoveries and advancements in research. Moreover, our findings demonstrate that researchers have employed numerous factors to measure Teamwork Quality (TWQ) and Teamwork Effectiveness (TWE) in ASD. Additionally, the observed similarity in questions across different instruments suggests the need for standardizing terminology. By highlighting the most frequent questions of each instrument, our results support the development of a unified Teamwork instrument in ASD.

The presented results offer valuable insights for both practitioners and researchers. For practitioners, this paper serves as a practical guide in utilizing the presented teamwork instruments, as it provides detailed information about their characteristics. This facilitates their practical application in Agile Software Development (ASD) projects. For researchers, this work highlights identified gaps and specialized factors specific to the agile context, offering opportunities to develop new instruments and advance the understanding of teamwork in ASD.

Future research endeavors should focus on establishing a unified taxonomy for teamwork instrument factors in ASD, creating a standardized framework to categorize and organize these factors consistently. Conducting longitudinal research can provide valuable insights into the evolution and effectiveness of teamwork instruments over time, enhancing our understanding of their performance in various contexts and identifying opportunities for adaptation and improvement. Additionally, investigating the relationship between specific teamwork instruments and project outcomes in ASD can shed light on how effective teamwork, as measured by these instruments, influences project success, productivity, and overall performance.

Supplementary Material

To ensure the study's transparency and completeness, we have provided a Supplementary Material [1] that contains the eight teamwork instruments factors, and questions. The additional methodological details and a comprehensive presentation of the results.

CRediT authorship contribution statement

Ramon Santos: Conceptualization of this study, Methodology, Software. **Felipe Cunha:** Data analysis, Writing - Review & Editing. **Thiago Rique:** Data collection, Visualization. **Mirko Perkusich:** Writing - Review & Editing. **Ademar Neto:** Supervision. **Danyllo Albuquerque:** Review & Editing. **Hyggo Almeida:** Funding acquisition, Resources. **Angelo Perkusich:** Project administration.

References

[1] Alderfer, C.P., 1983. An intergroup perspective on group dynamics. Technical Report. Yale Univ New Haven CT School of Organization and Management.

[2] Anderson, N., West, M.A., 1996. The team climate inventory: Development of the tci and its applications in teambuilding for innovativeness. European Journal of work and organizational psychology 5, 53–66.

[3] Armour, P.G., 2002. The spiritual life of projects. Communications of the ACM 45, 11–14.

[4] Dickinson, T.L., McIntyre, R.M., 1997. A conceptual framework for teamwork measurement, in: Team performance assessment and measurement. Psychology Press, pp. 31–56.

[5] Dikert, K., Paasivaara, M., Lassenius, C., 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software 119, 87–108.

[6] Dingsøyr, T., Fægri, T.E., Dybå, T., Haugset, B., Lindsjørn, Y., 2016. Team performance in software development: research results versus agile principles. IEEE software 33, 106–110.

[7] Freire, A., Neto, M., Perkusich, M., Gorgônio, K., Almeida, H., Perkusich, A., 2021. Towards a comprehensive understanding of agile teamwork: A literature-based thematic network, SEKE.

[8] Freire, A., Perkusich, M., Saraiva, R., Almeida, H., Perkusich, A., 2018. A bayesian networks-based approach to assess and improve the teamwork quality of agile teams. Information and Software Technology 100, 119–132.

[9] Goncalves, E., Lima, P., Cerdeiral, C., Diirr, B., Santos, G., 2021. Tact: An instrument to assess the organizational climate of agile teams-a preliminary study. Journal of Software Engineering Research and Development 9, 18–1.

[10] Gren, L., Torkar, R., Feldt, R., 2015. Group maturity and agility, are they connected?–a survey study, in: 2015 41st Euromicro Conference on Software Engineering and Advanced Applications, IEEE. pp. 1–8.

[11] Guzzo, R.A., Shea, G.P., 1992. Group performance and intergroup relations in organizations. .

[1] https://figshare.com/s/4b7ad201c314512f372b

[12] Hackman, J., 1987. The design of work teams. in jw lorsch (ed.), handbook of organizational behavior, englewood cliffs, nj: Prentice-hall .

[13] Hoegl, M., Gemuenden, H.G., 2001. Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. Organization science 12, 435–449.

[14] Lei, P.W., Wu, Q., 2007. Introduction to structural equation modeling: Issues and practical considerations. Educational Measurement: issues and practice 26, 33–43.

[15] Lindsjørn, Y., Sjøberg, D.I., Dingsøyr, T., Bergersen, G.R., Dybå, T., 2016. Teamwork quality and project success in software development: A survey of agile development teams. Journal of Systems and Software 122, 274–286.

[16] Lukusa, L., Geeling, S., Lusinga, S., Rivett, U., 2020. Teamwork and project success in agile software development methods: A case study in higher education, in: Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality, pp. 885–891.

[17] Marsicano, G., da Silva, F.Q., Seaman, C.B., Adaid-Castro, B.G., 2020. The teamwork process antecedents (tpa) questionnaire: developing and validating a comprehensive measure for assessing antecedents of teamwork process quality. Empirical Software Engineering 25, 3928–3976.

[18] Mathieu, J., Maynard, M.T., Rapp, T., Gilson, L., 2008. Team effectiveness 1997-2007: A review of recent advancements and a glimpse into the future. Journal of management 34, 410–476.

[19] Melo, C.d.O., Cruzes, D.S., Kon, F., Conradi, R., 2013. Interpretative case studies on agile team productivity and management. Information and Software Technology 55, 412–427.

[20] Moe, N.B., Dingsøyr, T., Dybå, T., 2010. A teamwork model for understanding an agile team: A case study of a scrum project. Information and software technology 52, 480–491.

[21] Moe, N.B., Dingsøyr, T., Røyrvik, E.A., 2009. Putting agile teamwork to the test–an preliminary instrument for empirically assessing and improving agile software development, in: Agile Processes in Software Engineering and Extreme Programming: 10th International Conference, XP 2009, Pula, Sardinia, Italy, May 25-29, 2009. Proceedings 10, Springer. pp. 114–123.

[22] Poth, A., Kottke, M., Riel, A., 2020. Evaluation of agile team work quality, in: Agile Processes in Software Engineering and Extreme Programming–Workshops: XP 2020 Workshops, Copenhagen, Denmark, June 8–12, 2020, Revised Selected Papers 21, Springer. pp. 101–110.

[23] Radhakrishnan, A., Zaveri, J., David, D., Davis, J.S., 2022. The impact of project team characteristics and client collaboration on project agility and project success: An empirical study. European Management Journal 40, 758–777.

[24] Russo, D., Stol, K.J., 2021. Pls-sem for software engineering research: An introduction and survey. ACM Computing Surveys (CSUR) 54, 1–38.

[25] Salas, E., Sims, D.E., Burke, C.S., 2005. Is there a "big five" in teamwork? Small group research 36, 555–599.

[26] Santos, R., Cunha, F., Rique, T., Perkusich, M., Almeida, H., Perkusich, A., Costa, Í., . A comparative analysis of agile teamwork quality instruments in agile software development: A qualitative approach .

[27] Serrador, P., Pinto, J.K., 2015. Does agile work?—a quantitative analysis of agile project success. International journal of project management 33, 1040–1051.

[28] Silva, M., Perkusich, M., Freire, A., Albuquerque, D., Gorgônio, K.C., Almeida, H., Perkusich, A., Guimaraes, E., 2022. A comparative analysis of agile teamwork quality measurement models. Journal of Communications Software and Systems 18, 153–164.

[29] Stavru, S., 2014. A critical examination of recent industrial surveys on agile method usage. Journal of Systems and Software 94, 87–97.

[30] Strode, D., Dingsøyr, T., Lindsjorn, Y., 2022. A teamwork effectiveness model for agile software development. Empirical Software Engineering 27, 56.

[31] Truong, D., Jitbaipoon, T., 2016. How can agile methodologies be used to enhance the success of information technology projects? International Journal of Information Technology Project Management (IJITPM) 7, 1–16.

[32] Van Assen, M.F., 2000. Agile-based competence management: the relation between agile manufacturing and time-based competence management. International Journal of Agile Management Systems 2, 142–155.

[33] Verwijs, C., Russo, D., 2023. A theory of scrum team effectiveness. ACM Transactions on Software Engineering and Methodology .

[34] Wheelan, S.A., Hochberger, J.M., 1996. Validation studies of the group development questionnaire. Small group research 27, 143–170.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

Directional Residual Frame: Turns the motion information into a static RGB frame

Pengfei Qiu[a], Yang Zou[a,*], Xiaoqin Zeng[a] and Xiangchen Wu[a]

[a]*Hohai University, China*

ARTICLE INFO

Article History:
Submitted 4.25.2023
Revised 7.31.2023
Accepted 8.5.2023

Keywords:
Motion information
Action recognition
Temporal difference module
DRF

ABSTRACT

The most commonly adopted methods of video action recognition are optical flow and 3D convolution. Optical flow method requires calculation in advance and a lot of computing resources. 3D convolution method encounters several problems such as many parameters, difficult training, and redundant computation. This paper proposes an approach that can turn the motion information into a static RGB frame by a feasible way of compression: Directional Residual Frame (DRF). This idea comes from a static cartoon that can represent complex events through residual shadows. DRF takes advantage of the scarce nature of residual frames in space and pixel value to achieve similar effects of residual shadows by fusing multiple residual frames. With the DRF, the motion information can be learnt as simply and efficiently as learning the RGB information. In addition, it also proposes a Short-term Residual Shadow Module based on the DRF. Experimental results show that it has better performance than the state-of-the-art model TDN on UCF101 benchmark..

1. Introduction

In recent years, Video-based action recognition has drawn a significant amount of attention from the academic community. In action recognition, there are two kinds of key and complementary information: appearances and motion. CNN have achieved great success in classifying images of objects, scenes, and complex events. Thus, it is crucial for action recognition to capture motion information in video, which is usually achieved by two kinds of mechanisms in the current deep learning approaches: two-stream network [1] and 3D convolutions [5,6,7]. Even though the two-stream network can effectively improve the accuracy of action recognition through the optical flow, it requires a lot of computing resources to extract the optical flow. Although the 3D convolution can learn motion features directly from the RGB frames, it also leads to large network models and high computational cost.

Therefore, how to efficiently learn motion information has been a crucial challenge in action recognition.

*Corresponding author
Email address: yzou@hhu.edu.cn(Y. Zou)

In everyday life, we can know the motion information of the meteor, the fan and other things through the residual shadow. Obviously, we acquire the motion information from a certain moment of spatial information. Think about it the other way. Can we use a 2D frame to characterize a complex movement process? The optical flow can only reflect the speed, and requires multiple pieces to characterize the non-linear motion. Comics are a very successful case in point. A cartoon can represent a wonderful fight by using a residual shadow. The shadow in static cartoons can be well characterized in the complex motion process. And temporal derivative (difference) is highly relevant to optical flow [2], and has shown effectiveness in action recognition by using RGB difference as an approximate motion representation [3, 4].

In this paper, we propose a motion representation approach based on RGB difference, termed as Directional Residual Frame (DRF). The principle of DRF is similar to the shadow in comics that turns the motion information into a static RGB frame. First, we subtract two adjacent frames in the video with absolute value to obtain residual frames [11]. Then, the residual frames are binarized. During the binarization process, the motion features are retained and the difference

caused by the brightness change is removed. Finally, the adjacent residual frames are fused to form a residual shadow-like trajectory map. As shown in Figure 1, our DRF is a good representation of the trees moving to the right (the movement caused by the camera movement) and the people running to the left.

Figure 1: The first 5 frames are consecutive frames in the video, and the sixth frame is the corresponding DRF.

To demonstrate the effectiveness of the DRF, we performed the experimental analysis using Temporal Difference Network (TDN) [12] on the benchmark UCF101 [13], which is the state-of-the-art method without optical flow and 3D convolution. TDN is able to yield a state-of-the-art performance on both motion relevant Something-Something V1 datasets [9] and scene relevant Kinetics datasets [10], under the setting of using similar backbones[12].

The technical contributions of the paper are summarized as follows:

- To reduce the serious redundant calculation in video understanding, we propose an effective compression approach DRF that can turn the motion information into a static RGB information by using the scarcity of residual frame, due to the high similarity between adjacent frames. Optical flow requires multiple stacks to react non-linear motion, whereas DRF demands only one.
- Based on the DRF, we propose a Short-term Residual Shadow Module (S-RSM) to capture the motion information.
- The experimental results show that compared with the S-TDM in the state-of-the-art model TDN, our approach achieves higher accuracy with fewer model parameters.

The rest of the paper is organized as follows. Section 2 proposes the concept and calculation process of the DRF, and presents the S-RSM module based on the DRF; Section 3 describes the details of the experiments and evaluates the effectiveness of our method on UCF101 benchmark; and Section 4 concludes the paper.

2. Directional Residual Frame

In this section, we describe the proposed DRF in detail. First, we give an overview of DRF. Then, we elaborate the calculation process of the DRF. Finally, we provide the implementation details of using DRF in TDN.

2.1 Overview

Residual shadow is the most successful case that turns RGB information into the motion information. Residual shadow has both motion trajectory information and direction information. So how to form a residual shadow from continuous frames is a challenge. Our thoughts of addressing this include two steps, as follows:

First, motion detection. In this step, the motion information is extracted from the sequential RGB frame. Objects undergoing spatial position changes in the image sequence are presented as foreground (motion region).

Second, motion fusion. In this step, the motion information extracted from the previous step is fused into a static RGB frame where the motion region blurs with time, like residual shadow.

Motion detection. The common methods for motion detection are: background subtraction, temporal difference and optical flow [17]. Both background subtraction and optical flow require a lot of computing resources, which are contrary to efficiency. Therefore, we adopt the temporal difference method to extract the motion object. The temporal difference method may mistakenly detect the area originally covered by the object as moving, called Ghost, which is a problem with motion detection. As shown in Figure 2, the area originally covered by the moving object will be incorrectly detected into motion, which is the Ghost. But Ghost will not be a problem here, because it can be used effectively in motion fusion.

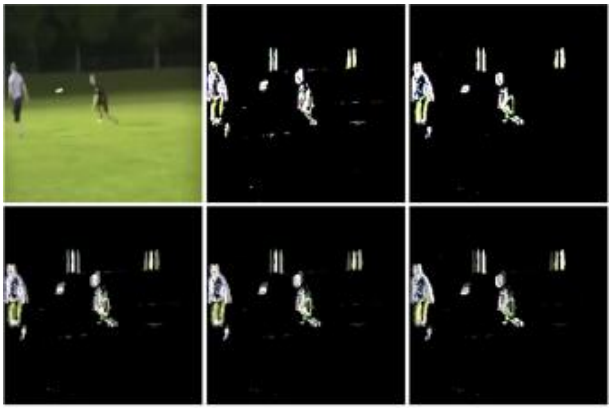


Figure 2: Motion Detection. The second and third frames are the two consecutive temporal differences before the first RGB frame. The fourth frame is *(df1 + df2)*, the fifth is *(2 * df2 + df1)*, and the sixth is the DRF, where df1 is Frame 2, and df2 is Frame 3.

Motion fusion. Motion fusion is the core of the proposed approach.

How to represent the direction of the movement is a crucial issue. In Figure 2, although the fourth and fifth frame preserve more motion traces with the scarcity of the residual frame, there is not any temporal information (direction information). The direction of motion is recognized with the aid of the numerical growth direction. In DRF, objects move from dark to light. From the sixth frame in Figure 2, it can be easily seen through residual shadow that the trees are moving to the right.

$$\begin{bmatrix}0&1&0&0\\1&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}+2*\begin{bmatrix}0&1&0&1\\1&0&0&1\\0&0&0&0\\0&0&0&1\end{bmatrix}+2^2*\begin{bmatrix}1&0&0&1\\1&0&1&0\\0&1&0&0\\0&0&0&0\end{bmatrix}=\begin{bmatrix}4&3&0&6\\7&0&5&2\\0&5&0&0\\0&0&0&3\end{bmatrix}$$

Figure 3: Binary fusion. Every matrix of 5 * 5 represents a residual frame, and the region of the matrix with an element value of 1 indicates the foreground.

Another issue is how to preserve the complete information in the motion fusion process. Although the residual frame is scarce, the foreground of different residual frames may overlap. The overlapping region of the foreground of two adjacent residual frames is the Ghost of the latter residual frame. As for more than two frames, the overlapping region will become difficult to interpret. The overcoverage approach, where the overlapping region takes the same value as the late residual frame, would lose a lot of information. Our approach is inspired by the binary coding to use the value-domain scarce nature of the residual frame. Each number of pixel values indicates an overlapping possibility. In Figure 3, the region with an element value of 7 in the resulting matrix is the overlapping region of 3 matrices; and the region with a value of 5 is the overlapping region of the first and third matrices.

2.2 The calculation process of the DRF

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

The process of calculating the RDF is divided into three steps. First, Temporal difference is employed to remove background and acquire motion region. Then, the binarization is adopted to remove the noise and obtain the scarce residual frame. Finally, the DRF is obtained from the fusion of residual frames.

Step 1: Residual frames

Residual frames contain more motion-specific features by removing still objects and background information and leaving mainly the changes between frames [11]. As shown in the third frame in Figure 4, the movement region will be brighter than the static areas.

The movement regions in the residual frame achieve positive or negative values, which are highly correlated with the pixel value of the background. The correlation can cause the movement regions being either positive or negative in the residual frame, thus failing to know the direction of the object. In Figure 4, the Frisbee is white with values above the background color, so it moves from the negative to the positive area in the residual frame. But this direction of movement is unreliable. Therefore, we utilize the absolute residual frame to alleviate the interference of the pixel value of the background. The issue of motion direction in the absolute residual frame will be tackled in step 3.

Figure 4: The first three frames are adjacent frames, the fourth one is the corresponding residual frame, the fifth one is the residual frame after the absolute value, and the sixth one is a binarization of the fourth one.

Here we use $Frame_i$ to represent the i_{th} frame data, and $Frame_{i\sim j}$ denotes the stacked frames from the i_{th} frame to the j_{th} frame. The process of obtaining residual frames can be formulated as follows:

$$ResFrame_{i\sim j}=\left|Frame_{i\sim j}-Frame_{i+1\sim j+1}\right|$$

At this stage, the Residual frames is not a sparse matrix. Influenced by the camera motion and light intensity changes, the gray area is not all *0*.

Step 2: Binarization

Binarization of the residual frames: *0* is used to represent no change area, and *1* is used to represent change area. In the field of image segmentation, there are a few algorithms [14] for image binarization. In this paper, we adopt threshold method in order to reduce the amount of computation as much as possible.

The formula is as follows:

$$threshold=\frac{\alpha}{n^2}*\sum_{x=1}^{n}\sum_{y=1}^{n}ResFrame_i(x,y)+\beta \quad (1)$$

$$ResFrameB_i(x,y)=\begin{cases}1 & ResFrame_i(x,y)>threshold\\ 0 & otherwise\end{cases} \quad (2)$$

Here ResFramei(x,y) is the image value of the

coordinates (x, y) in the ith residual frame. α and β are super parameters. And n is the size of the i_{th} residual frame.

a. Threshold with $\alpha = 1$ and $\beta = 0$

b. Threshold with $\alpha = 1$ and $\beta = 0.05$

Figure 5: Binarization. The "mean" in the images represents the mean of the residual frames. The first chart of each row is the pixel value statistics chart, the second is the residual frame, and the third is the binarized residual frame.

Since residual frame is a scarcity matrix, the mean value tends to be below the minimum in the movement region. With very small amplitude of motion, the mean may be lower than the difference value caused by changes in light intensity. We denote the minimum of the threshold by β. Figure 5 illustrates the effect of β on removing the background noise.

Step 3: Motion fusion. The higher the value is, the later the event occurs.

The motion fusion of multiple binary residual frames transforms temporal information into numerical information. The higher the value is, the later the event occurs.

$$DRFrame_n^{'} = \sum_{i=1}^{n} 2^{i-1} * ResFrameB_i \tag{3}$$

We accumulate the residual frames according to formula 3. There may be overlaps between the differences of consecutive residual frames. Various overlapping cases of n residual frames will be mapped to the value $0 \sim 2^n$. The case with n=4 is shown in Figure 6. In Figure 3, the region in the result matrix corresponding to the motion region (value is 1) in the third matrix should obtain the maximum value to indicate the movement end point. But the value of overlapping region will be greater than the last motion region. The brightest region appears in the middle region of the motion trajectory, as in the fifth frame of Figure 2.

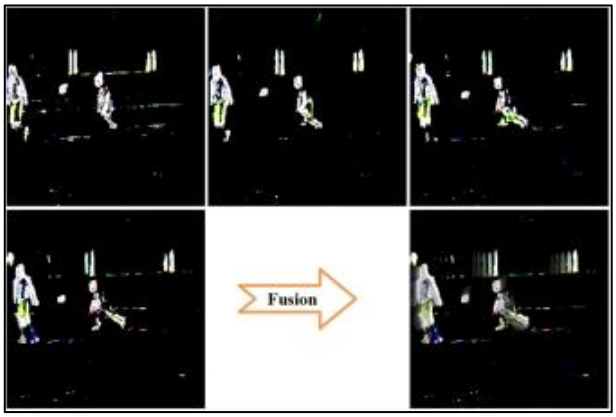


Figure 6: The first four pictures are continuous residual frames after binarization, and the last one is the DRF fused by the first four.

$$Mask_i(x,y) = \begin{cases} 1 & DRFrame_n^{'}(x,y) = 2^{i-1} \\ 0 & otherwise \end{cases} \tag{4}$$

$$DRFame_n = DRFame_n^{'} + \sum_{i=1}^{n} 2^{i-1}\left(ResFrameB_i * Mask\right) \tag{5}$$

In Formula 4, the operator sets the non-zero element in the matrix to *0* and the zero element to *1*. Then, through Formula 5, the value of the non-overlapping difference is doubled.

2.3 S-RSM with DRF

Based on the DRF, a Short-term Residual Shadow Module (S-RSM) is proposed, as an improvement of the S-TDM in TDN, as illustrated in Figure 8.

Temporal Difference Networks (TDN) is a video-level architecture of capturing both short-term and long-term information for end-to-end action recognition. TDN is composed of a short-term and long-term temporal difference module (TDM), as illustrated in Figure 7 [12]. In Figure 9, the short-term TDM in TDN supply a single RGB frame with a temporal difference to yield an efficient video representation, explicitly encoding both appearance and motion information [12].

In TDN, the stacks of difference frames are processed by 2D convolution, which can only capture limited motion information and of which the main function is to calibrate the moving area on the static image.

The DRF turns the action information into the static RGB information by fusing multiple temporal difference frames. So the model can capture the movement information by learning the RGB information in the DRF. This feature of DRF is beneficial to 2D convolutional networks to learn motion features, so as to perform the task of action recognition even better.

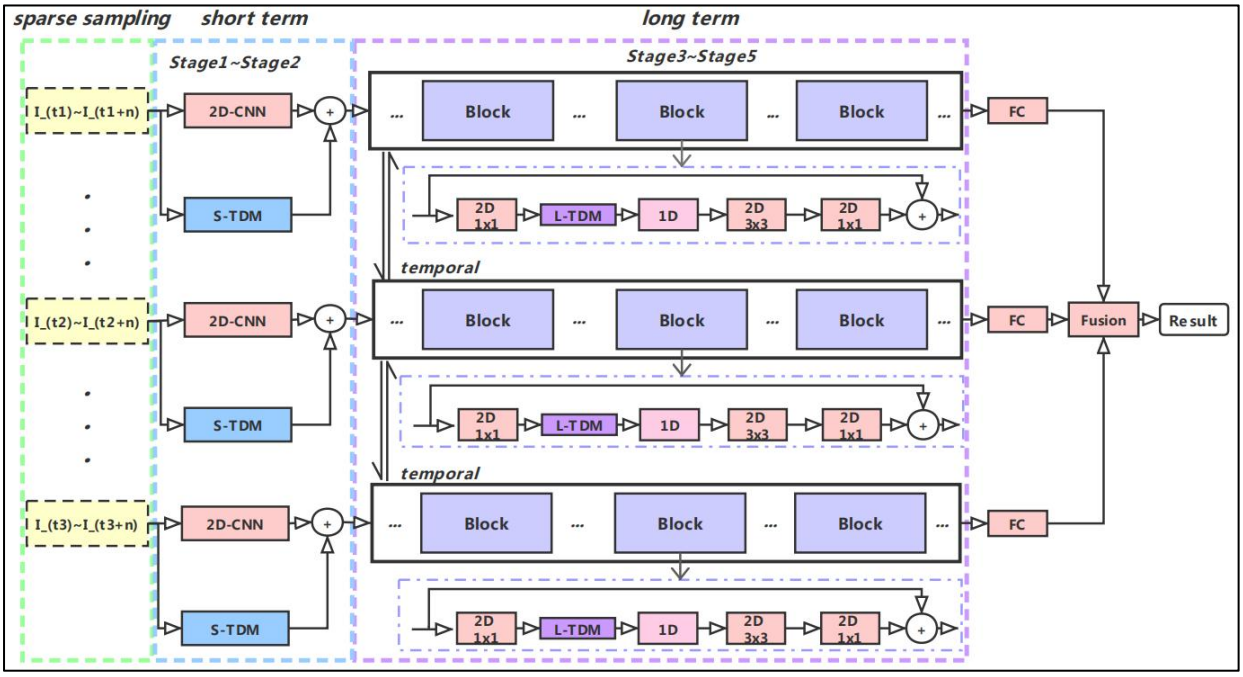


Figure 7: Framework of Temporal Difference Network (TDN). Based on the sparse sampling from multiple segments, TDN aims to model both short-term and long-term motion information.

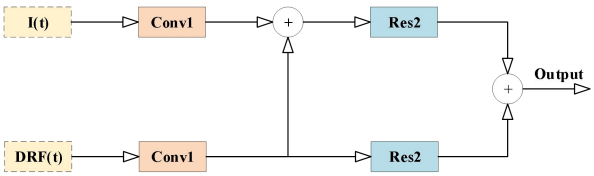


Figure 8: Framework of short-term Residual Shadow Module with DRF

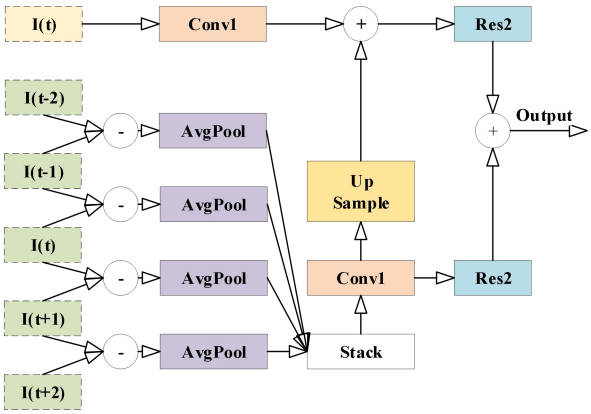


Figure 9: Framework of the short-term TDM.

3. Experiments

In this section, we present the experiment results of the proposed DRF. First, we describe the evaluation datasets and implementation details. Then, we evaluated the effectiveness of DRF on the state-of-the-art method TDN.

3.1 Datasets and implementation details

Video datasets. There are several commonly used datasets for video recognition tasks. We mainly focus on the benchmark: UCF101. UCF101 consist of 13,320 videos in 101 action categories [13].

Training and testing. In experiments, we use ResNet50 to implement TDN framework. Following common practice [15], during training, each video frame is resized to have shorter side in [256, 320] and a crop of 244 × 244 is randomly cropped. We pre-train TDN on the ImageNet dataset [16]. The batch size is 128 and the initial learning rate is 0.02. The total training epoch is set to 60 in the UCF101 benchmark. The learning rate will be divided by a factor of 10 when the performance on validation set saturates. For testing, the shorter side of each video is resized to 256. We implement the kind of testing scheme: 1-clip and center-crop where only 1 center crop of 244 × 244 from a single clip is used for evaluation.

3.2 Experimental Results

From the experimental results in Table I, it can be found that the beta of DRF taking 0.05 is a suitable value. The binarized threshold as the mean has lower accuracy than the other two schemes, which confirms the viewpoint we mentioned in Section 2. With very small amplitude of motion, the mean may be lower than the difference value caused by changes in light intensity.

Since the residual frame is absolute, the beta of DRF is the lower limit of the threshold. The accuracy of the beta of DRF being 0.05 is higher than that of the beta of DRF being 0.1. When the threshold is set too high,

some important motion information will be filtered out.

Table 1: ACC of different binarization parameters on UCF101 benchmark

Method	Backbone	Input (S-RSM)	Length (DRF)	Alpha (DRF)	Beta (DRF)	Top-1
TDN	ResNet50	DRF	5	1.00	0.00	84.51%
TDN	ResNet50	DRF	5	1.00	0.05	85.33%
TDN	ResNet50	DRF	5	1.00	0.10	84.92%

From the experimental results in Table II, it can be shown that the frames of the DRF motion fusion is of length 5 in UCF101 benchmark. When the DRF length is set to 3, the reason for the decrease of accuracy is that TDN learns too little action information, whereas it is set to 7 and 9, the reason for the decrease of accuracy is that it is difficult for TDN to learn.

Table 2: ACC of different motion fusion length on UCF101 benchmark

Method	Backbone	Input (S-RSM)	Length (DRF)	Alpha (DRF)	Beta (DRF)	Top-1
TDN	ResNet50	DRF	3	1.00	0.05	84.29%
TDN	ResNet50	DRF	5	1.00	0.05	85.33%
TDN	ResNet50	DRF	7	1.00	0.05	84.48%
TDN	ResNet50	DRF	9	1.00	0.05	84.48%

The results in Table III show that the proposed TDN outperforms the original model at sampling frames of 4 and 8. With the sample frame of 4, our approach improves by nearly 1% over the original method; and with the sample frame of 8, our approach improves by more than 1.2%.

From this set of comparative experiments, it can be concluded that DRF contains better motion information than the stacked residual frames. In [12], it has been shown that TDN can reach the state-of-the-art level without the use of optical flow and 3D convolution.

Table 3: ACC of different module on UCF101 benchmark

Method	Backbone	Input	module	Frames	Top-1
TDN (original)	ResNet50	RGB + difference	S-TDM	4	84.97%
TDN (original)	ResNet50	RGB + difference	S-TDM	8	87.15%
TDN (ours)	ResNet50	RGB + DRF	S-RSM	4	85.95%
TDN (ours)	ResNet50	RGB + DRF	S-RSM	8	88.39%

4. Conclusion

To address the problem of serious redundant calculation in video motion recognition, we propose the approach to squeezing the motion information into a RGB frame. The principle of DRF is similar to the shadow in comics. The shadow in static cartoons can be well characterized in the complex motion process. DRF exploits the scarcity of residual maps to fuse the motion information of multiple residual maps into one spatial frame. In this way, it can learn motion information as it learns about RGB information. Based on the DRF, we propose S-RSM based on 2D convolution to capture motion information. Through comparative experiments, we verified that our approach has better performance than the state-of-the-art model TDN in UCF101 benchmark.

References

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in NIPS, 2014, pp. 568-576.

[2] Berthold K.P. Horn and Brian G. Schunck, "Determining optical flow," Artificial Intelligence, vol.17, pp. 185-203, 1981.

[3] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. lin, X. Tang and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," European conference on computer vision, vol. 9912, pp. 20-36, 2016.

[4] Z. Yue, Y. Xiong, and D. Lin, "Recognize Actions by Disentangling Components of Dynamics," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 6566-6575.

[5] S. Ji, W. Xu, M. Yang and K. Yu, "3D convolutional neural networks for human action recognition." IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 35, pp. 221-231, 2013.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," Proceedings of the IEEE international conference on computer vision, 2015, pp. 4489-4497.

[7] K. Hara, H. Kataoka, and Y. Satoh, “Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018, pp. 6546–6555.

[8] J. Lin, C. Gan, and S. Han, "TSM: Temporal Shift Module for Efficient Video Understanding," IEEE/CVF International Conference on Computer Vision (ICCV) IEEE, 2019, pp. 7082-7092.

[9] J. Carreira, and A. Zisserman, "Quo vadis, action recognition? a new model and the Kinetics Dataset," Computer Vision and Pattern Recognition IEEE, 2017, pp. 4724-4733.

[10] R. Goyal, SE. Kahou, V. Michalski, J. Materzynska and R. Memisevic, "The" something something" video database for learning and evaluating visual common sense," Proceedings of the IEEE international conference on computer vision, 2017, pp. 5843-5851.

[11] L. Tao, X. Wang, and T. Yamasaki, "Rethinking motion representation: Residual frames with 3D ConvNets," IEEE Transactions on Image Processing, vol. 30, pp. 9231-9244, 2021.

[12] L. Wang, Z. Tong, B. Ji and G. Wu, "TDN: Temporal Difference Networks for Efficient Action Recognition," Computer Vision and Pattern Recognition IEEE, 2021, pp. 1895-1904.

[13] M. S. Hutchinson and V. N. Gadepally, "Video action understanding," IEEE Access, vol. 9, pp. 134611-134637, 2021.

[14] Otsu, N. "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems Man & Cybernetics, vol. 9, pp. 62-66, 2007.

[15] C. Feichtenhofer, H. Fan, J. Malik and K. He, "Slowfast networks for video recognition," Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6201-6210.

[16] L. J. Li, R. Socher and F. F. Li, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework," IEEE Conference on Computer Vision & Pattern Recognition IEEE, 2009, pp. 2036-2043.

[17] A. A. Shafie, F. Hafiz and M. H. Ali, "Motion detection techniques using optical flow," World Academy of Science Engineering & Technology, 2009

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

GraPH: Graph Partitioning Based on Hotspots

Hiba G. Fareed[a], Isam A. Alobaidi[b,c,*], Jennifer L. Leopold[d] and Andrea E. Smith[d]

[a]Mathematics Department, Mustansiriyah University, Baghdad, Iraq
[b]School of Computer Science and Information Systems, Northwest Missouri State University, Maryville, MO, USA
[c]Department of Computer Engineering, Al Farabi University College, Baghdad, Iraq
[d]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA

ARTICLE INFO

Article History:
Submitted 10.25.2023
Revised 11.5.2023
Accepted 12.5.2023

Keywords:
graph partitioning
graph data mining
structures
hotspot.

ABSTRACT

Graphs have long been used to model relationships between entities. For some applications, a single graph is sufficient; for other problems, a collection of graphs may be more appropriate to represent the underlying data. Many contemporary problem domains, for which graphs are an ideal data model, contain an enormous amount of data (e.g., social networks). Hence, researchers frequently employ parallelized or distributed processing. But first the graph data must be partitioned and assigned to the multiple processors in such a way that the work load will be balanced, and inter-processor communication will be minimized. The latter problem may be complicated by the existence of edges between vertices in a graph that have been assigned to different processors. Herein we introduce a strategy that combines vocabulary-based summarization of graphs (VoG) and detection of hotspots (i.e., vertices of high degree) to determine how a single undirected graph should be partitioned to optimize multi-processor load balancing and minimize the number of edges that exist between the partitioned subgraphs. We benchmark our method against another well-known partitioning algorithm ($METIS$) to demonstrate the benefits of our approach.

1. Introduction

Graphs are frequently used as an abstraction to model the real-world data; such as in chemical, or biological networks. The nature of the application problem will determine whether these data will be represented in a single or a collection of graphs. This diversity has contributed to the proper representation of the underlying data. Some of these graphs may contain an enormous amount of data (e.g., social networks). Hence, parallelized or distributed processing often is employed. Before the analysis commences, typically the graph dataset is partitioned, and a subset of data is assigned to each processor. The partitioning should be done in such a way that the ensuing work load will be balanced and inter-processor communication will be minimized. These tasks can be particularly challenging for a single graph; consideration must be given to which vertices are assigned to which partitions (i.e., processors) and what edges originally existed between those vertices.

Ideally, partitions should be of approximately equal size, and the number of edges between vertices that are in different partitions should be minimized. The problem of finding good partitions in these respects has been studied in graph theory. Despite the numerous algorithms that have been proposed and implemented, the complexity of this problem is still considered NP-complete.

In general, most graph partitioning algorithms utilize either edge-cut partitioning or vertex-cut partitioning. Edge-cut partitioning splits the vertices of a graph into disjoint sets of approximately equal size considering the minimum number of cut-edges (e.g., PowerGraph [3], Spark GraphX [4], and Chaos [14]). In contrast, vertex-cut partitioning splits the edges of a graph into equal-sized sets. In this approach, the partitioning of a single graph must satisfy two require-

*Corresponding author
hf_math@uomustansiriyah.edu.iq (H.G. Fareed); ialobaidi@nwmissouri.edu (I.A. Alobaidi); leopoldj@mst.edu (J.L. Leopold); aes7dc@mst.edu (A.E. Smith)
https://uomustansiriyah.edu.iq/e-learn/profile.php?id=5609 (H.G. Fareed); https://www.nwmissouri.edu/csis/directory/alobaidi.htm (I.A. Alobaidi); https://cs.mst.edu/people/faculty-directory/ (J.L. Leopold)
ORCID(s): 0000-0002-6508-2495 (H.G. Fareed); 0000-0001-6329-2440 (I.A. Alobaidi)

DOI reference number: 10-18293/JVLC2023-N2-039

ments: the quality graph partitioning criterion (which guarantees no lost data) and load balancing. Many studies have shown that edge-cut partitioning produces more accurate results on large real-world graphs [3, 4].

Herein we introduce a novel vertex-cut partitioning strategy that determines how a single, undirected graph should be partitioned to optimize multi-processor load balancing and minimize the number of edges that exist between the partitioned subgraphs. Our approach, $GraPH$, first uses vocabulary-based summarization [9] to identify the most highly connected structures that exist in the graph (e.g., cliques, stars, and chains). We then find the vertices in those structures that have the highest degree; these are called hotspots. The hotspots become the starting points from which subgraph partitions are formed.

This paper is organized as follows. In Section 2 we briefly discuss some of the related work in graph partitioning. We present the $GraPH$ algorithm in Section 3, and include a discussion of the VoG summarization algorithm. In Section 4 we experimentally evaluate our proposed algorithm ($GraPH$) to expound its benefits. Concluding remarks and a discussion of future work are provided in Section 5.

2. Related Work

In this section, we briefly review some of the research that has been done in graph partitioning. Despite the numerous sequential, distributed, and parallel algorithms that have been developed, the complexity of this problem is still considered to be NP-complete. One of the most significant challenges of the problem continues to be minimizing the loss of information (from the original graph dataset) when the partitions are formed; that is, the goal is to minimize the number of edges (from the original graph) that exists between vertices that are in different partitions, a situation which is more likely to occur as the number of partitions increases.

Some heuristic methods for sequential graph partitioning of a single graph are discussed in [6, 2]. One offline method (wherein the entire graph is resident in memory), $METIS$, is proposed in [6]. This method produces high-quality partitions in terms of uniformity of partition size and minimization of "lost" edges. However, because of the offline setting, it cannot handle large graphs. The $METIS$ algorithm consists of three phases: coarsening, partitioning, and refinement. During each phase, a sequence of specialized algorithms is applied. These algorithms help in selecting the maximal matchings in the coarsening phase, partitioning of the coarse graph in the partitioning phase, and projecting the graph back to the original graph in the refinement phase. An extension to $METIS$ (Streaming $METIS$ Partitioning method (SMP)) is proposed in [2], replacing the offline setting of $METIS$ by an online setting. SMP provides the ability to adjust the memory capacity, and subsequently decrease computational requirements by applying the partitioning method to small subgraphs.

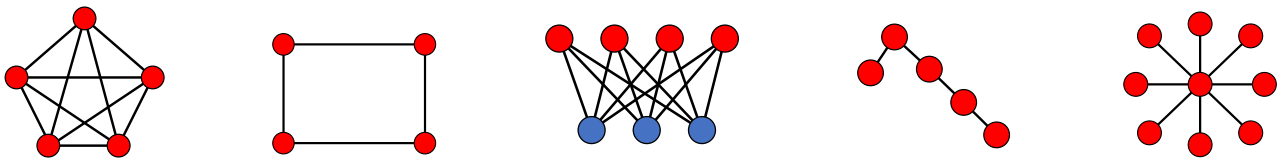
Some graph partitioning techniques are designed for specific application problems. Another technique for local (i.e., memory-resident, sequential processing) graph partitioning [1] specifically targets fixed cardinality problems such as k-densest subgraph and max k-vertex cover. The authors developed a fixed parameter algorithm using a greediness-for-parameterization technique. Clustering systems are used as a base in [17]. In this research, the authors propose a heuristic graph edge partitioning strategy, Neighbor Expansion (NE), with polynomial running time. Their goal was to reduce the running time and communication cost for some specific applications such as triangle counting and PageRank.

The graph partitioning problem in a distributed environment is addressed in [12, 11, 8, 15, 7]. The authors in [12] propose a fully distributed algorithm called JA-BE-JA. This algorithm is built on two types of partitioning: vertex-cut and edge-cut partitioning; the absence of central coordination and the processing of each vertex independently make this algorithm well-designed for distributed processing. Another distributed algorithm, $PACC$ (Partition-Aware Connected Components), based on graph partitioning for edge-filtering and load-balancing, is proposed in [11]. The authors of [15] propose a multi-level label propagation (MLP) method that uses distributed memory of several machines for partitioning the graphs. Another distributed partitioning algorithm is discussed in [10], PARallel Submodular Approximation algorithm ($Parsa$), also configures the partitions to fit the storage and computation ability of each machine.

One important characteristic of graph partitioning algorithms is the strategy employed for selecting the vertex around which the subgraph will be built for each partition. Many algorithms select such vertices randomly. Our approach was motivated by $MELT$ [16], MapReduce-based Efficient Large-scale Trajectory anonymization. The main objective of that work was to examine paths traveled by people in a geographical space, and then partition the space into regions around popular locations (e.g., a coffee house, an exercise center, etc.); those locations are referred to as hotspots. As will be discussed later in this paper, the utilization of hotspots as a basis for forming partitions is a novel feature of our partitioning strategy.

3. Methodology

In this section, we present the $GraPH$ strategy for partitioning a single, undirected graph. We begin with some preliminary definitions that will facilitate this discussion. An explanation of the vocabulary-based summarization of graphs (VoG) technique developed in [9] then follows; this is a key component for our approach as it is used to determine subgraphs of high connectivity (e.g., cliques, stars, and chains). Finally, our complete set of algorithms is presented, detailing how the vocabulary-based summarization and identification of hotspots lead to the creation of optimal partitioning.



(a) Full Clique. (b) Near Clique. (c) Full Bipartite. (d) Chain. (e) Star.

Figure 1: Types of Structures.

3.1. Preliminaries

Definition 1. ***Graph:*** *A graph G is a tuple (V, E, L) where V is a finite set of nodes called the vertex set of G, and E is a set of 2-element subsets of V ($E \subseteq V \times V$) called the edge set of G. The nodes and edges are labeled by the function L.*

Definition 2. ***Graph partitioning:*** *A graph $G = (V, E)$ will be partitioned into k subgraphs $G'_{sub} = (V', E')$, $sub = 1, ..., k$. Each $V'_{subset} \subset V_{set}$ where $V_i \cap V_j = 0$ for $i \neq j$, and each $E'_{subset} \subset E_{set}$.*

Definition 3. ***Full-clique:*** *Let $G = (V, E)$ be an undirected graph. A set FC of vertices in G is called a Full-clique if any two distinct vertices in FC are adjacent in G, when $k \geq 1$. The Full-clique term may refer to the subgraph in some cases. If several edges are missing, this will be defined as a* ***Near-clique****.*

Definition 4. ***Full bipartite core:*** *Let $G = (V, E)$ be an undirected graph. A set Fb of vertices in G is called Full-bipartite if two sets of vertices S_1 and S_2, $S_1 \cap S_2 = \emptyset$, have edges between them, where each vertex in S_1 will be connected to every edge in S_2 but not within the same set. When the core is not fully connected this will be defined as a* ***Near-bipartite core****.*

Definition 5. ***Star:*** *A* ***Star*** *consists of one internal vertex in set S_1 connected to k edges of other sets S_{i+1} (spokes). A Star is considered as a special case of a Full bipartite core.*

Definition 6. ***Chain:*** *A* ***Chain*** *is a sequence of vertices such that all vertices have degree 2, except two of them have degree 1.*

Figure 1 shows examples of these structure types.

3.2. *VoG* Graph Summarization

The ability to summarize information about highly connected subgraphs contained within a large graph can greatly facilitate understanding of the graph as a whole. Vocabulary-based summarization of Graphs (VoG) [9] is a formal methodology developed for this purpose. Using a set of terms (i.e., a vocabulary) like full-cliques, near-cliques, full-bipartite core, near-bipartite core, stars, and chains, VoG provides a summary of the most highly connected and frequently occurring structures in a graph. For problem domains like social networks and communication networks, these are typically the structures of most interest.

Algorithm 1 outlines the main steps that are performed in VoG; see [9] for a more detailed discussion. Using graph decomposition methods, candidate subgraphs are first generated. They are then classified as various connected structures such as cliques, stars, and chains; if a subgraph qualifies as more than one of these structure types, a scoring method (based on minimum description length (MDL)) is used to determine which structure type that subgraph best fits. VoG then uses another scoring system to determine which collection of those structures best characterizes the graph as a whole. This is called the summary model, and could include all of the structures (PLAIN), just the k structures with the best scores (TOP10, TOP100), or a combination of structures whose total score is best (GREEDY'nFORGET).

Algorithm 1 VoG

1: **Input** Graph G.
2: **Output** Graph summary M, encoding cost.
3: **Subgraph Generation.** Using graph decomposition methods, produce a set of candidate subgraphs, which may overlap with each other.
4: **Subgraph Labeling.** Characterize each subgraph as one of the vocabulary structure types.
5: **Summary Assembly.** From the candidate structures, select a non-redundant subset to instantiate the graph model M. Utilizing a heuristic model (e.g., PLAIN, TOP10, TOP100, GREEDY'nFORGET), the set of structures with the lowest description cost will be selected.

3.3. Proposed Algorithm

Two algorithms have been proposed here one dealing with Sequential processing while the other dealing with Parallel processing

3.3.1. Sequential Algorithm

In $GraPH$, we first use VoG to identify the most highly connected, and frequently occurring, subgraphs. That produces a set of structures (i.e., the model summary), S. Algorithm 2 is then used to select a subset of S which we call the majority structures, $MajS$. The number of majority structures depends on the desired number of partitions, n. The n structures in S that have the largest number of vertices become the majority structures.

For each majority structure, Algorithm 3 is applied to identify the vertex that has the highest degree; in the case of a tie, an arbitrary choice between those qualifying vertices is made. These vertices of highest degree are called hotspots.

Algorithm 2 Select the Majority Structures

Input S is set of structures produced by VoG,
  n is number of desired partitions
Output $MajS$ contains n structures in S that have the largest number of vertices
$SortedS$ = Sort structures in S in descending order by number of vertices;
  for $i = 1$ to n **do**
    $MajS[i] = SortedS[i]$
  end-for
return $MajS$

Algorithm 3 Assign the HotSpot

Input $S = (V_S, E_S)$ is a structure
Output $HotSpot$ is a vertex in V_S that is the hotspot vertex for structure $S = (V_S, E_S)$
  for $i = 1$ to $|V_S|$ **do**
    $degree[i] = 0$
  end-for
  for $i = 1$ to $|V_S|$ **do**
    for $j = 1$ to $|V_S|$ **do**
      if there is an $edge(i, j)$ in E_S
        then $degree[i] = degree[i] + 1$
      end-if
    end-for
  end-for
  $HotSpot = 1$
  for $i = 2$ to $|V_S|$ **do**
    if $degree[HotSpot] <= degree[i]$
      then $HotSpot = i$
    end-if
  end-for
return $HotSpot$

After assigning the hotspots, the actual partitioning commences. The subgraph that will be assigned to a partition will consist of all the vertices in a hotspot's structure unless that number of vertices exceeds the total number of vertices in the graph divided by the number of desired partitions; that is considered the ideal partition size. In Algorithm 4, we start a depth-first search from a hotspot vertex (denoted as Hotspot). The $MajS$ denoted in the algorithm is the set of structures from which the hotspot was selected. There are two discontinuation criteria for building a subgraph partition; the expansion will stop when either of those conditions is satisfied:

1. The current size of a partition subgraph has reached the ideal partition size.
2. The path length from the current vertex to the hotspot has reached a maximum threshold (i.e., the total number of desired partitions).

Some vertices from the original graph may not be included in any partition using these conditions. To handle those cases, we perform a breadth-first search starting from each hotspot until all nodes are included in some partition.

Algorithm 4 GraPH

Input Graph $G = (V, E)$ and $HotSpot$ and $MajS$
  $MajS$ is a set contains structures that have the largest number of vertices
  $HotSpot$ is a vertex in the structure connected to the largest number of edges
  n is the number of partitions
Output All $SubGraph$s of G, where $|V|$ of each subgraph $\geq PartitionSize$
$PartitionSize = |V| \ / \ n$
  if $|MajS_i| \leq PartitionSize$ **then**
  Include all nodes of $MajS_i$ in $Partition_i$
  end-if
  Perform DFS starting from each $HotSpot$
  $SubGraph_{DFS} \leftarrow DFS$ result
  Perform BFS starting from each $HotSpot$
  $SubGraph_{BFS} \leftarrow BFS$ result
$SubGraph \leftarrow SubGraph_{DFS} \cup SubGraph_{BFS}$
return $SubGraph$

3.4. Computational Complexity

The complexity of one well-known partitioning method that is considered to produce high-quality partitions, $METIS$ [6] (implemented as kmetis), is approximately $O(V + E + k\log(k))$ where V is the number of nodes, E the number of edges, and k is the number of partitions [5]. In contrast, the complexity of $GraPH$ is approximately $O(V + E + n\log(n))$ where V is the number of nodes, E is the number of edges, and n is the number of structures. Contributing to the overall complexity of $GraPH$ is the complexity of BFS and DFS, which are $O(V + E)$, and the complexity of sorting n structures, which is $O(nlog(n))$. We are not including the complexity of the VoG processing, which has not been published by its authors.

4. Results and Analysis

In this section we compare the results of partitioning three datasets using $GraPH$ and another well-known partitioning method, $METIS$, which was discussed in Section 2. The $GraPH$ algorithms presented in Section 3.2 and 3.3.1 were (collectively) implemented in Matlab and C++. A C++ implementation of $METIS$ was downloaded from the Karypis Lab website [5]. Our experiments were executed on an Intel(R) Core(TM) i7-6700 CPU@3.40GHz computer with 32 GB memory.

4.1. Data Description

Three single undirected graphs were used to evaluate our approach. Table 1 lists descriptive information about the graphs. One graph was synthetically generated; a second graph represented a two-dimensional finite element mesh; the third graph represented a three-dimensional finite element mesh. The last three graphs were obtained from the Network Repository,a large comprehensive collection of network graph data [13].

Table 1
Description of the Graphs Tested

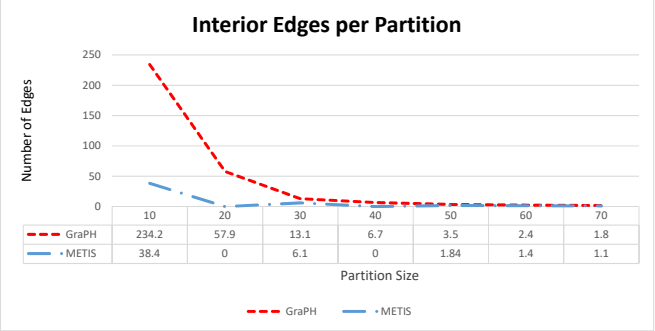
Graph Name	Number of Nodes	Number of Edges	Description
Synthetic	1565	3561	Synthetically generated
4ELT	15606	45878	2D Finite element mesh
COPTER2	55476	352238	3D Finite element mesh
web-wikipedia-link-fr	4.9M	113.1M	Power-Law
road-road-usa	23.9M	28.8M	Low-Degree
soc-sinaweibo	58.6M	261.3M	Long-Tailed

4.2. Experiment and Results

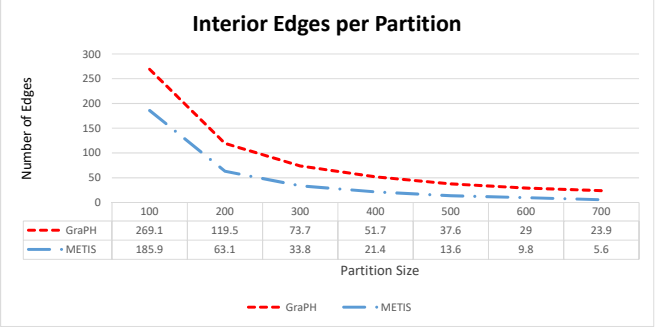
We executed *GraPH* and *METIS* on each of the graphs listed in Table 1, testing seven different numbers of partitions for each graph. The results from each test were analyzed in terms of three different metrics: the number of *interior edges* per partition (i.e., edges in a partition's graph), the number of *exterior edges* per partition (i.e., edges between vertices in a partition and vertices assigned to other partitions), and the total number of *edges lost* (i.e., edges from the original graph that were not represented in any of the partition graphs).

Seven tests were conducted to create 10, 20, 30, 40, 50, 60, and 70 partitions, respectively, of the Synthetic graph. *METIS* failed to partition this graph into either 20 or 40 partitions; the program simply failed to return any results. *GraPH* produced results for all of the tested numbers of partitions for this graph. The representation of edges amongst partitions was not well distributed when 10 partitions were requested. Specifically, the number of interior edges in one of those partitions was much higher than in the other partitions, which was not an optimal partitioning. This was likely due to the fact that when a hotspot is selected from a structure, if the structure can fit entirely into a partition, all nodes from that structure automatically will be added to the partition before the depth-first search algorithm is run. This can then prevent other partitions from growing during depth-first search (as would be the case in unconnected components), encouraging disproportionate partition sizes.

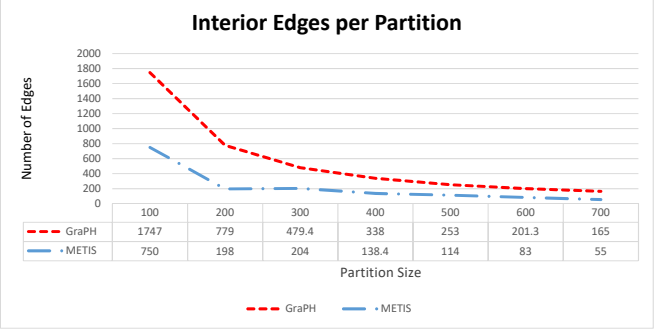
Because the 4ELT and COPTER2 graphs were much larger than the Synthetic graph, we tested larger numbers of partitions for those graphs, namely: 100, 200, 300, 400, 500, 600, and 700. For all three of the graphs listed in Table 1, in the majority of the tests, the partitions produced by *GraPH* had a higher number of interior edges in each partition than the partitions produced by *METIS*. It can be seen in Figure 2 that more edges from the original graph were retained within the partitions produced by



(a) 1565 Nodes - 3561 Edges.



(b) 15606 Nodes - 45878 Edges.



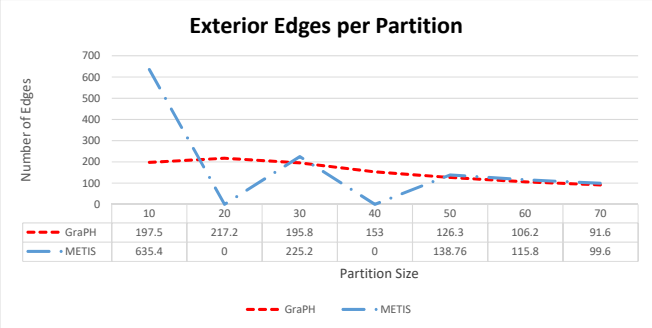
(c) 55476 Nodes - 352238 Edges.

Figure 2: Interior Edges per Partition.

GraPH. As shown in Figure 3, the *GraPH* partitioning resulted in fewer exterior edges (between partitions) than what occurred in the *METIS* partitioning. Additionally, as shown in Figure 4, *GraPH* outperformed *METIS* in terms of reducing the total number of edges lost from the original graph. It should be noted that as the desired number of partitions grew, the difference in partition quality (in terms of the three metrics) obtained from both methods became less distinct.

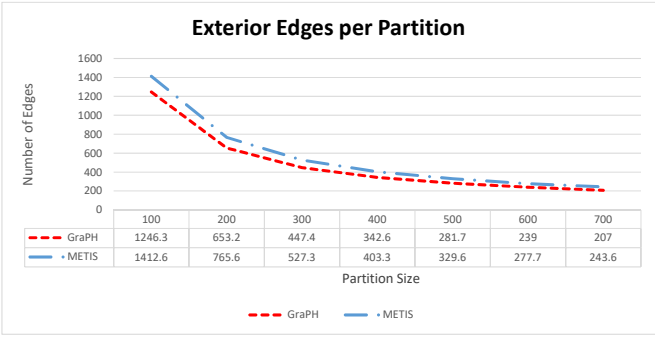
Because of the use of two methods (depth-first/breadth-first search) in *GraPH* for the extension process that include vertices in/out of partition boundaries, we also evaluated different variations of our method. We ran *GraPH* on the three test graphs using four different orders of processing:

- Depth-first search extension for vertices inside the partition boundaries followed by breadth-first search ex-



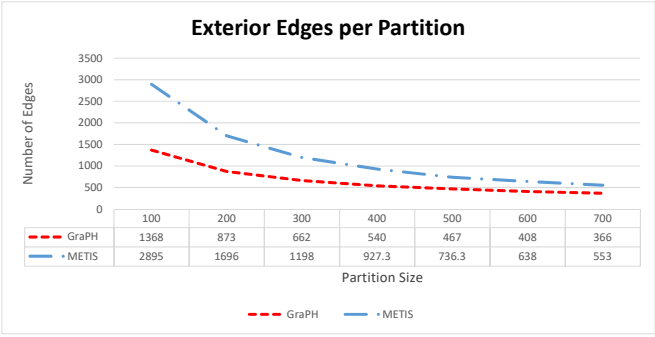
	10	20	30	40	50	60	70
GraPH	197.5	217.2	195.8	153	126.3	106.2	91.6
METIS	635.4	0	225.2	0	138.76	115.8	99.6

(a) 1565 Nodes - 3561 Edges.



	100	200	300	400	500	600	700
GraPH	1246.3	653.2	447.4	342.6	281.7	239	207
METIS	1412.6	765.6	527.3	403.3	329.6	277.7	243.6

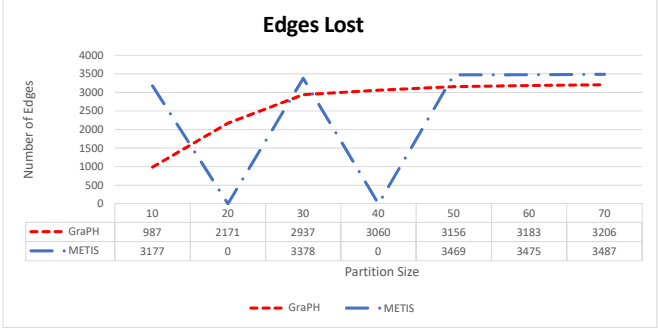
(b) 15606 Nodes - 45878 Edges.



	100	200	300	400	500	600	700
GraPH	1368	873	662	540	467	408	366
METIS	2895	1696	1198	927.3	736.3	638	553

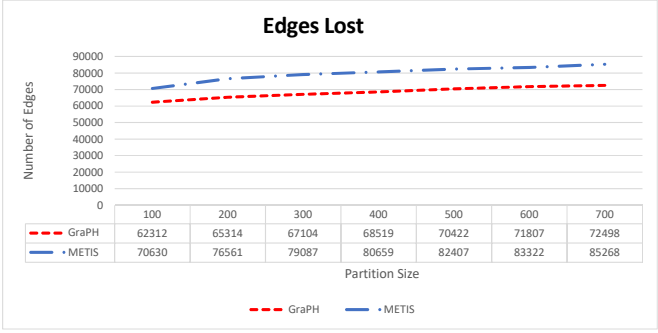
(c) 55476 Nodes - 352238 Edges.

Figure 3: Exterior Edges per Partition.



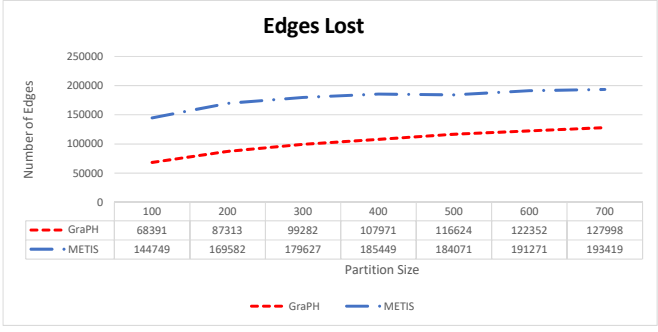
	10	20	30	40	50	60	70
GraPH	987	2171	2937	3060	3156	3183	3206
METIS	3177	0	3378	0	3469	3475	3487

(a) 1565 Nodes - 3561 Edges.



	100	200	300	400	500	600	700
GraPH	62312	65314	67104	68519	70422	71807	72498
METIS	70630	76561	79087	80659	82407	83322	85268

(b) 15606 Nodes - 45878 Edges.



	100	200	300	400	500	600	700
GraPH	68391	87313	99282	107971	116624	122352	127998
METIS	144749	169582	179627	185449	184071	191271	193419

(c) 55476 Nodes - 352238 Edges.

Figure 4: Total Edges Lost.

tension for vertices outside the partition boundaries.

- Breadth-first search extension for vertices inside the partition boundaries followed by depth-first search extension for vertices outside the partition boundaries.
- Depth-first search extension for vertices inside the partition boundaries followed by depth-first search extension for vertices outside the partition boundaries.
- Breadth-first search extension for vertices inside the partition boundaries followed by breadth-first search extension for vertices outside the partition boundaries.

We found that more consistent partitions were obtained (in terms of more interior edges and fewer external edges per partition) when we utilized the depth-first search extension process for vertices inside the boundaries followed by breadth-first search extension processing for vertices outside the boundaries. We also tested random assignment of hotspots. This was found to be unreliable in generating high-quality partitions. Interestingly, although the number of internal edges was not balanced across partitions utilizing randomization, $GraPH$ still outperformed $METIS$ in terms of producing partitions with more internal edges and fewer external edges.

5. Conclusion and Future work

With the proliferation of data in our technological world and the usefulness of modeling some problems using graphs, it is becoming increasingly difficult to process an entire graph dataset in memory. It is more efficient to partition a single large graph, and process multiple smaller subgraphs. However, in doing so, the partitioning of what may be highly interconnected data must be done in such

as way as to balance the work load amongst the individual processes, minimize inter-process communication, and minimize loss of information from the original dataset. The latter problems can occur if, in the original graph, there is an edge that exists between vertices assigned to different partitions.

Herein we have presented an algorithm, $GraPH$, for partitioning a single, undirected graph. Our algorithm strives to produce quality partitions in terms of: uniformity of the size of each partition, maximization of the number of edges from the original graph that are included in each partition, and minimization of the number of edges from the original graph that effectively exist between partitions. Our approach is novel; we first utilize vocabulary-based summarization (VoG) to find the most highly connected structures, and then find the vertices of highest degree (known as hotspots) within those structures. A benchmark comparison of $GraPH$ with another well-known, high-quality partitioning algorithm ($METIS$) demonstrated the benefits of our strategy.

In the future, we plan to explore ways to distribute or parallelize the $GraPH$ algorithms so that we can process even larger graphs than those tested for this study. To that end, we also may explore the use of some approximation (e.g., sampling) methods that may increase the efficiency of the assignment of vertices to partitions after identification of structures and hotspots.

6. Acknowledgments

The authors thank Dr. Danai Koutra for her assistance in executing the VoG software.

References

[1] Bonnet, É., Escoffier, B., Paschos, V.T., Tourniaire, É., 2015. Multi-parameter Analysis for Local Graph Partitioning Problems: Using Greediness for Parameterization. Algorithmica 71, 566–580. URL: https://doi.org/10.1007/s00453-014-9920-6, doi:10.1007/s00453-014-9920-6.

[2] Echbarthi, G., Kheddouci, H., 2016. Streaming METIS Partitioning, in: Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '16), IEEE Press, Piscataway, NJ, USA. pp. 17–24. URL: http://dl.acm.org/citation.cfm?id=3192424.3192429, doi:10.1109/ASONAM.2016.7752208.

[3] Gonzalez, J.E., Low, Y., Gu, H., Bickson, D., Guestrin, C., 2012. Powergraph: Distributed Graph-parallel Computation on Natural Graphs, in: Proceedings of the 10th. USENIX Symposium on Operating Systems Design and Implementation (OSDI '12), USENIX Association, Hollywood, CA, USA. pp. 17–30.

[4] Gonzalez, J.E., Xin, R.S., Dave, A., Crankshaw, D., Franklin, M.J., Stoica, I., 2014. Graphx: Graph Processing in a Distributed Dataflow Framework, in: Proceedings of the 11th. USENIX Symposium on Operating Systems Design and Implementation (OSDI '14), USENIX Association, Broomfield, CO, USA. pp. 599–613.

[5] Karypis, G., 2007. Complexity of pmetis and kmetis Algorithms. http://glaros.dtc.umn.edu/gkhome/node/419. Accessed: 2019-22-01.

[6] Karypis, G., Kumar, V., 1998a. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. SIAM Journal on Scientific Computing 20, 359–392. URL: https://doi.org/10.1137/S1064827595287997, doi:10.1137/S1064827595287997.

[7] Karypis, G., Kumar, V., 1998b. A Parallel Algorithm for Multilevel Graph Partitioning and Sparse Matrix Ordering. Journal of Parallel and Distributed Computing 48, 71–95. URL: http://www.sciencedirect.com/science/article/pii/S0743731597914039, doi:https://doi.org/10.1006/jpdc.1997.1403.

[8] Kiveris, R., Lattanzi, S., Mirrokni, V., Rastogi, V., Vassilvitskii, S., 2014. Connected Components in Mapreduce and Beyond, in: Proceedings of the ACM Symposium on Cloud Computing (SOCC '14), ACM, New York, NY, USA. pp. 18:1–18:13. URL: http://doi.acm.org/10.1145/2670979.2670997, doi:10.1145/2670979.2670997.

[9] Koutra, D., Kang, U., Vreeken, J., Faloutsos, C., 2015. Summarizing and Understanding Large Graphs. Statistical Analysis and Data Mining: The ASA Data Science Journal 8, 183–202. doi:10.1002/sam.11267.

[10] Li, M., Andersen, D.G., Smola, A.J., 2015. Graph Partitioning via Parallel Submodular Approximation to Accelerate Distributed Machine Learning. CoRR 1505.04636. URL: http://arxiv.org/abs/1505.04636.

[11] Park, H.M., Park, N., Myaeng, S.H., Kang, U., 2016. Partition Aware Connected Component Computation in Distributed Systems, in: Proceedings of the 16th. IEEE International Conference on Data Mining (ICDM '16), IEEE. pp. 420–429. doi:10.1109/ICDM.2016.0053.

[12] Rahimian, F., Payberah, A.H., Girdzijauskas, S., Jelasity, M., Haridi, S., 2015. A Distributed Algorithm for Large-Scale Graph Partitioning. ACM Trans. Auton. Adapt. Syst. 10, 12:1–12:24. URL: http://doi.acm.org/10.1145/2714568, doi:10.1145/2714568.

[13] Rossi, R., Ahmed, N., 2015. The network data repository with interactive graph analytics and visualization. Proceedings of the AAAI Conference on Artificial Intelligence 29. URL: https://ojs.aaai.org/index.php/AAAI/article/view/9277.

[14] Roy, A., Bindschaedler, L., Malicevic, J., Zwaenepoel, W., 2015. Chaos: Scale-out Graph Processing from Secondary Storage, in: Proceedings of the 25th. Symposium on Operating Systems Principles (SOSP '15), ACM, New York, NY, USA. pp. 410–424. URL: http://doi.acm.org/10.1145/2815400.2815408, doi:10.1145/2815400.2815408.

[15] Wang, L., Xiao, Y., Shao, B., Wang, H., 2014. How to Partition a Billion-Node Graph, in: Proceedings of the 30th. IEEE International Conference on Data Engineering (ICDE '14), IEEE. pp. 568–579. doi:10.1109/ICDE.2014.6816682.

[16] Ward, K., Lin, D., Madria, S., 2017. MELT: Mapreduce-based Efficient Large-scale Trajectory Anonymization, in: Proceedings of the 29th. International Conference on Scientific and Statistical Database Management (SSDBM '17), ACM, New York, NY, USA. pp. 35:1–35:6. URL: http://doi.acm.org/10.1145/3085504.3085581, doi:10.1145/3085504.3085581.

[17] Zhang, C., Wei, F., Liu, Q., Tang, Z.G., Li, Z., 2017. Graph edge partitioning via neighborhood heuristic, in: Proceedings of the 23rd. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17), ACM, New York, NY, USA. pp. 605–614. URL: http://doi.acm.org/10.1145/3097983.3098033, doi:10.1145/3097983.3098033.

Journal of Visual Language and Computing

Volume 2023, Number 2