Bachelor Computer Science & Economics

Digitizing human expertise in process mining

Renske van Gulijk (s3291480)

Supervisors: Peter van der Putten, Aske Plaat

Bachelor Thesis Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl 01-07-2025

Abstract Process mining is a technique used to improve complex, large-scale industry processes by analysing company data in the form of event logs. The event log analysis can, for example, help to identify risks and inefficiencies, or can be used to create a solutions methodology for process improvement. Although human experts often perform process mining, this research will improve upon an agentic setup from previous work, whereby one human expert is mimicked by a digital Domain Knowledge Gathering Agent (DKGA) to answer process risk and inefficiency questions.

In this research, the main experiment demonstrated that introducing specialization through role prompting within multiple DKGAs significantly improved the output for answering process risk and inefficiency questions, compared to a setup that used a single general-purpose DKGA. Furthermore, a setup with three DKGAs outperformed a setup with five because integrating outputs from a larger number of agents increased the complexity and made it more difficult to combine the information accurately.

A human-in-the-loop experiment provided the DKGAs with the human trait of interaction by allowing the user of the system to provide feedback, resulting in increased user-friendliness and explainability of the setups. Despite these traits, the quality of the written output did not improve significantly.

The most promising future work, which builds upon this research, is automating the creation of role-specialized DKGAs based on the complexity of (process risks and inefficiency) questions.

Keywords: process mining · multi-agent systems · domain knowledge · CrewAI framework

Contents

1 Introduction

It is widely accepted in macroeconomics that, in a society, specialisation is almost always more economically efficient. This research investigates whether the same is true for digital cooperating agents in the field of process mining. For this, digital knowledge engineering - a "branch of AI that attempts to mimic the judgment and behavior of a human expert" (Vomberg et al., 2024) - is used.

Process mining is a technique used to improve complex, large-scale industry processes of businesses and organizations by analysing event log data. The technique creates model abstractions from these event logs to, for example, identify risks and develop solutions methodologies to improve processes. A process mining analysis traditionally relies on domain experts' opinions to supply domain knowledge, because they can justify the reasoning for risks or solutions. Hiring multiple domain experts can result in substantial resource investments for product owners.

This research improved upon a previous agentic setup by Vogt et al. (2024), which used a single agent to gather external domain knowledge and create abstractions from internal domain knowledge to answer process risk and process inefficiency questions. The internal domain knowledge files are company-specific files that have been made public, mostly for competitions in process mining. The setup used a planner agent to create sub-questions, a gathering agent to collect domain knowledge, and a writer agent that combined all the knowledge to answer the questions. A user feedback interaction was mentioned in the future work.

In this research, human experts are mimicked by digital Domain Knowledge Gathering Agents (DKGAs) in two ways. The first is the introduction of more DKGAs, which are specialized through role prompting to become domain experts. Creating different domain experts introduces different views of the same abstraction. The second is introducing an interaction between the user and the agents. In a human setting, the user would interact with the experts to provide feedback or address different requirements. The main research question is: *To what extent do interactive and specialized DKGAs improve responses to process risk and inefficiency questions compared to a single general agent*? The main experiment will increase the amount of DKGAs to create an expertise specialization, and a human-in-the-loop experiment takes the human trait of interaction by providing a user feedback integration method.

The main experiment consisted of three setups: one DKGA (DoS0, the baseline), three DKGAs (DoS2), and five DKGAs (DoS4). The Degree of Specialization (DoS) refers to the number of DKGAs minus one. This is because a setup with a single DKGA is a general agent and does not have a role specialization; therefore, the DoS is zero.

The human-in-the-loop experiment tested two DoS2 setups that implemented the human trait of interaction by allowing the user of the system to provide feedback to the DKGAs. One setup, DoS2CH, had a within-agent method in which the user was prompted for feedback at the end of every DKGA run. The other setup, DoS2FH, implemented a between-agent feedback method in which the user was able to provide feedback based on the final report. The DoS2FH setup would use the feedback to rerun a certain agent and recombine the results.

In the agentic setups, the DKGAs work together to gather external domain knowledge, which, in combination with internal domain knowledge, is used to answer process mining questions. External domain knowledge refers to knowledge that is available and accessible (on the internet) in contrast to internal domain knowledge, which is private company data. An agent can be understood as a Large Language Model (LLM) with a certain role, operating in a collaborative agentic environment. It has the property of being able to use agentic tools.

The architecture and framework choices for the experiment setups are selected with two agentic patterns and two agentic concepts as criteria. They are: automatic state sharing, independent concurrency between agents, user feedback processing, and agent resilience. All three setups in the main experiment (DoS0, DoS2, DoS3) and the DoS2CH from the human-in-the-loop experiment use a sequential multi-agent system architecture and a CrewAI Crew framework. This adheres to three of the four agentic patterns and concepts. The agentic pattern of "independent concurrency between agents" is restricted due to the sequential running of the agents. On the other hand, the DoS2FH setup uses a goal-based modular architecture and a CrewAI Flow framework. This setup adheres to only one of the agentic patterns and concepts, namely the "user feedback processing". However, using additional programming, a pre-programmed class instance introduces part of the shared state functionality.

The five setups from the main and the human-in-the-loop experiments are tested on six process risk and process inefficiency questions. These six questions have a different focus, company, and process. Three examples for the focus are: audit risks, process inefficiencies, and cyber risks. Companies that have been used in the questions are, for example, Google, Volvo, and Wells Fargo Bank. A selection of processes that have been analysed are the loan application process, the order-to-cash process, and the accounts payable process.

Three evaluation methods are used within this research: measuring the contribution percentage, a statistical Wilcoxon rank test, and a references analysis. The 'contribution percentage' is a measure of whether all of the DKGAs contribute to answering the question in every paragraph. In the main experiment, the contribution percentage is calculated per question and a statistical Wilcoxon rank test is performed. In the human-in-the-loop experiment, the contribution percentage is also calculated per question and a reference analysis is performed.

The structure of this thesis is as follows. Section 2 covers the knowledge needed to understand the creation of DKGAs and which architectures and frameworks are available that match agentic patterns and concepts. Following this, Section 3 covers five similar works. Two works make use of single DKGA creation in Process Mining, one work introduced DKGA specialization using different training instead of role-prompting, and two works implemented specialization through DKGA role-prompting. With this gathered information, Section 4 covers the methodological choices that were made to answer the main research question. This includes the question design, an explanation of how DKGAs are specialized, a selection of tools, frameworks, architectures, and the evaluation method. Using the selected methods, the experimental setup in Section 5 explains clearly how the two experiments were conducted. The results, in Section 6, summarize all the findings from the experiments. The discussion, Section 7, analyzes these findings, summarizes the limitations, and provides future research areas. Section 8 concludes the paper. The Appendix contains a side study that was conducted to select the evaluation methods of this research.

2 Background

Process mining is a technique that can be used to improve a complex and large-scale processes within businesses and organizations. This is done by analysing company-specific big data such as communications, company payments, and planning documents in the form of large event logs or user interactions (Bhaskar & Osama, 2025). Three applications of process mining are, for example: the creation of (textual or visual) abstractions from event logs, identifying risks, and creating a

solutions methodology for process inefficiencies.

For process mining applications, knowledge from different domains is required. Traditionally, this is gathered by employing experts in specific domains. This research will use interactive role-specialized digital agents instead of human experts, meaning that a larger audience can be reached than traditional methods: product owners who do not possess the resources for a process mining analysis with human experts.

2.1 Why domain knowledge is important to process mining

Due to the unavailability of resources, process mining is often performed by a single domain expert. Abstaining from much-needed domain knowledge can make process mining unreliable because process components can be misinterpreted, leading to an inaccurate risk analysis or solutions methodology to improve processes (Zerbato et al., 2022). This misinterpretation is often caused by jargon (Zimmermann et al., 2024).

A theoretical example of why domain knowledge is important to process mining is as follows. In a flour factory, the process of packaging might be very different from that in a computer factory. If a process mining analysis is done on the event logs of the computer factory, the abstractions might show that the packaging step is inefficient because it is slow, with a lot of repetition. With added domain knowledge, it becomes clear that processes in this area are slow because a lot of careful computer checking is done by hand, and that the quality is more important than the quantity in this domain. Domain knowledge may also be used to find potential risks or opportunities in the process, such as worker conditions that need to adhere to the EU regulations, and the sustainability of the packaging material used.

2.2 Creating DKGAs

A study by Vomberg et al. (2024) stated that there is currently a development in AI "from narrow applications to a versatile general-purpose technology". This improves the functionality of DKGAs because they can be adapted to act as experts across different domains due to the amount of external knowledge available on the internet.

To be able to create DKGAs, there are several components that need to be pre-defined: a clear role definition, a task, and available tools. The role and the task definition, covered in Section 2.2.1, ensure that the agent has a well-defined specialization, and agent output. It is possible to create agents without a task, however, because DKGAs optimally should gather "accessible, useful, reliable, and ethical" (Vomberg et al., 2024) information, assigned tasks can ensure correct functionality.

Tools are "predefined code paths" (Malfa et al., 2025) that can help to increase the correct execution of that specialization. The agent is in control of when these tools are called, but they cannot change anything about the execution of the code. Some general rules-of-thumb for tools are covered in Section 2.2.2, these rules will be taken into account in the methodology.

Often, DKGAs have a web-scraping tool to gather domain knowledge. The reason for specific knowledge gathering tools is that the LLMs behind the agents are prone to hallucinate facts (Ji et al., 2023) and have the "inability to access up-to-date information" (Chen et al., 2024). For these reasons, the DKGAs in this research will be provided with tools for the domain knowledge gathering ability. Available options are covered in background section 2.2.3.

1. 'The Physicist'

Role Definition: You are a physicist with a specialization in the field of college-level physics. Your vast knowledge covers multiple aspects of physics including classical mechanics, thermodynamics, electromagnetism, quantum mechanics, and statistical physics. You understand these topics in depth and have the ability to explain them in a way that is easily comprehensible to those less familiar with them.

Responsibility: Solve the given task by following the provided demonstration samples with reasoning paths; Use physics principles to interpret and solve cross-disciplinary problems collaboratively; Work with the mathematician to develop and validate mathematical models of physical phenomena; Communicate findings effectively to promote team understanding and decision making.

Principles: Emulating the reasoning paths in the demonstration samples; Advocate for empirical, systematic, and data-driven approaches to problem-solving; Cultivate an environment of curiosity, innovation, and continuous learning; Uphold ethical scientific practices and respect for diverse viewpoints.

Figure 1: DKGA with a physicist specialization

Within this research, a tool is needed to create an abstraction from an event log. One DKGA will perform this action to create an understandable internal domain knowledge abstraction. All DKGAs may use the created abstraction as internal domain knowledge and combine it with external domain knowledge to analyse it. To create the abstractions, three options are given in background section 2.2.4.

2.2.1 Assigning DKGA specialization using role prompting

The idea of agent specialization is not new. A survey from 2024 (Ling et al., 2024) stated that it was an "emerging field of study with [...] substantial potential for impact". For the DKGAs in this research, a specialization will be assigned using prompt engineering with role-prompting techniques.

Prompt engineering is the study of how to correctly instruct agents to be able to ensure a correct execution. Role-prompting is a prompting method that gives the agent a specific role to operate in, which encourages the agent to answer the task according to a specific role-based expectation. Correctly-implemented role prompting helps to improve DKGA performance (Kong et al., 2024). A research by Q. Wu et al. (2023) concluded that the agent gained "more effective consideration of [...] conversation context and role alignment" compared to a traditional "task-style prompt". The task-style prompt may astray the agent from correct output contexts.

Agent role prompting often has three requirements: a role definition, a specific domain task, and an explanation of how to respond within its role. An example of these three requirements is shown in Figure 1, where the DKGAs role is a physics teacher (Chen et al., 2024).

A theoretical example of creating a DKGA with a role specialization is as follows. For the previously mentioned flour-producing factory, a DKGA with a specialization in production machines may be created. If the role states clearly that it is a "research expert in the domain of production machines for a flour factory," it is more likely that the agent returns a written paper with references to scientific papers while sticking to the domain context. Using task-style prompting may lead the agent to search in other domains or write less formally, despite the fact that it may be stated in the task-based prompt.

Role-prompting, like all other prompting, needs to be well-thought-out. Research from 2023 established that the overused role-specialization of a "helpful assistant" (Zheng et al., 2023) produced significantly worse outputs than an agent with a more specialized role-prompt.

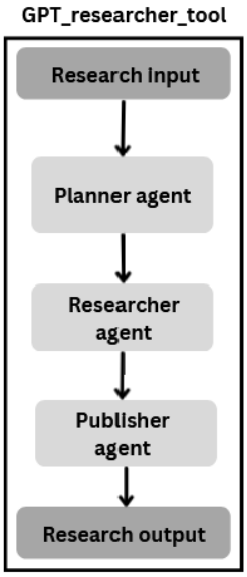

Figure 2: GPT_researcher agents interaction

2.2.2 General rules for the selection of tools

The choice of tool selection may greatly impact the final result. Two general rules for the selection of tools will be covered. The first is that tools should be selected carefully based on documentation and research papers that make use of these tools. The second is that an agent should not possess too many tools.

A research from 2024, which varied the tools of an LLM (Matthew Lam et al., 2024) showed that selection could lead to a maximum performance variation of 50%. The selection of tools needs to be done with careful consideration.

Another general rule for tool selection is that the agents should not possess too many tools because it could lead to selecting the wrong tools. Traditionally, in agentic frameworks, models are provided with a large number of tools. However, one of the main conclusions of research done by J. Wu et al. (2025) was that having just two tools (a web search and a coding tool) was sufficient to perform even the most difficult assigned tasks; they called their finding the "less is more" principle.

2.2.3 Tools for the DKGA's domain knowledge gathering ability

There are many tools available to help the DKGAs gather up-to-date and accurate information. From all available tools, a small selection will be considered because they are simple to implement, used in research, and free to use. These are: DuckDuckGo Search, Tavily search, GPT_researcher, and Bing Web Search. All of these tools use API calls to connect with their search engine. The API call returns the top few search results and a summary of the content.

DuckDuckGo search[1] has the main advantage of being "privacy-conscious" (Saravanos et al., 2022). In research done by Zhang et al. (2023), the tool was used to gather content from "online community websites" to help programmers with writing code.

[1]https://pypi.org/project/duckduckgo-search/

Tavily search[2] is widely used in academia due to its simplicity, and free usage. It "[delivers] real-time, accurate, and factual results at speed"[3]. A selection of research in computer science that uses this tool is: creating coding helper agents (Soni et al., 2025), optimizing high-performance computing coding (Rahman et al., 2025), and safety testing of LLMs (Ugarte et al., 2025).

The Bing web search[4] created by Microsoft has the same core functionality as the Tavily search. It uses the Bing Web Search API to find information. A research implementation from 2022 used the tool to retrieve information online to answer questions in a web browser environment (Nakano et al., 2022). One downside is that Microsoft will be removing the functionality as of August 2025[5].

The GPT_researcher tool[6] is a multi-agent system that can be called as a tool in certain frameworks. It includes a task agent that receives the task, a planner that divides the task into executable research questions, a researcher that retrieves references, and a publisher that writes the paper. GPT_researcher makes use of the OpenAI GPT-4o-mini model and the Tavily search tool. It works with coroutines, which are implemented with the asynchio module [7]. Coroutines ensure that several elements of the code can run in parallel, such as several research executions.

2.2.4 Tools for the DKGA's internal domain knowledge analysis

The first step in process mining is to understand the event log of the process that is being analysed. The event log is supplied as part of internal domain knowledge. One DKGA needs a specialization in process mining to be able to create an event log abstraction through process discovery. Using the abstraction, the DKGA can gather external domain knowledge to select the most important elements of the abstraction, which will be shared with the other agents.

Three tools are considered for the event log abstraction, namely the PM4PY, Apromore, and ProM tools. All of these tools can do process discovery, conformance checking, and performance analysis. For this background research, only the process discovery is relevant.

PM4PY (Process Mining for Python) is an open-source Python tool. It has various process discovery modules, such as a directly-follows graph (DFG), which discovers the frequencies in the event log and the temporal profile, which discovers time differences in data. The documentation is very thorough[8] and a scientific paper by Berti et al. (2023) provides an "in-depth overview" of the library. There are also many scientific works which use the tool for process mining, two examples use the PM4PY process discovery for analysing event logs (Tripathi et al., 2024; Vogt et al., 2024).

Apromore is a web-based process mining service that can be called through a specially made API call called REST API. In terms of desired functionalities for this research, it is the same as PM4PY. The documentation is also very thorough for this tool[9]. A scientific application is a research by Conforti et al. (2015), which uses Apromore to analyse "business process variants".

The ProM[10] tool can also adhere to the directly-follows graph, variants, and temporal profile process discovery. An explanatory research about ProM done by van Dongen et al. (2005) explained

[2]https://github.com/tavily-ai/tavily-python
[3]https://python.langchain.com/docs/integrations/tools/tavily_search/
[4]https://www.microsoft.com/en-us/bing/apis/bing-web-search-api
[5]https://learn.microsoft.com/en-us/bing/search-apis/bing-web-search/overview
[6]https://github.com/assafelovic/GPT-researcher
[7]https://docs.python.org/3/library/asyncio.html
[8]https://pm4py-source.readthedocs.io/en/latest/
[9]https://documentation.apromore.org/
[10]https://promtools.org/

that ProM integrates the functionality of several existing process mining tools, and that the coding is a "pluggable" environment. The tool was introduced by a scientific research from 2005 (van Dongen et al., 2005).

A comparison paper of PM4PY and ProM by Gomes et al. (2022) concluded that PM4PY outperformed ProM for process discovery in terms of customization and costs.

2.3 Environment for DKGAs: architectures and frameworks

Two agentic patterns and two agentic concepts are chosen as criteria for selecting available agentic environments in this research. The two agentic patterns are "automatic state sharing" and "independent concurrency between agents". The two agentic concepts are: 'user feedback processing" and 'agent resilience". In this paper, agentic concepts are qualities that an agent can possess, while agentic patterns show how an agent works or behaves to reach its goals.

The two agentic patterns and two agentic concepts are based on four agentic patterns from a work by Singh et al. (2024). The four agentic patterns in the research found are: "reflection", "multi-agent collaboration", "planning", and "tool utilization". Of these four agentic patterns, two will be rephrased into agentic concepts, one will be split into two specific patterns, and the last will be disregarded.

The provided definition of the "reflection" agentic pattern is "allowing systems to analyze and improve their outputs iteratively" (Singh et al., 2024). This research will implement a human-in-the-loop, in which the user provides feedback. The agentic pattern "reflection" will be changed to the agentic concept of "user feedback processing". This agentic concept clears up that the reflection is due to a user input, and not that the agents are able to improve the outputs themselves.

The "planning" agentic pattern is defined as "the strategic formulation of actions to achieve specific goals" (Singh et al., 2024). Since this is closely linked to how well the agents communicate and interact, a more specific agentic concept of "agent resilience" is needed because the setup might have to withstand unforeseen crashes or errors in individual agents. This agentic concept will check whether the agents are able to rerun themselves with different prompting or be skipped without making the setup crash.

"Multi-agent collaboration" as an agentic pattern was defined by Singh et al. (2024) as "systems to perform in concert towards common objectives". Multi-agent collaboration is an optional agentic pattern that will be a useful criterion for this research. To provide more focused criteria, this pattern will be broken down into the two agentic patterns of "automatic state sharing" and "independent concurrency between agents".

If the agents can automatically share states, they can collaborate through shared communication. Agents can function without having a shared state; however, it can help to steer the agent's behaviour to better fulfil its task.

The agentic pattern of "independent concurrency between agents" will be taken into consideration when selecting an architecture and a framework. The agentic pattern is inspired by a critical research paper by Malfa et al. (2025), which stated that agentic systems often fall short in this area. In other words, it identified a common difference between theory and implementation: "LLM-agents often operate in strictly sequential concurrency pipelines or parallel, rather than as independent, concurrently operating agents".

The agentic pattern of "tool utilization" will not be considered as a framework criterion because all considered frameworks already support tool use.

The four criteria are the agentic concepts of "user feedback processing" and "agent resilience", and the agentic patterns of "automatic state sharing" and "independent concurrency between agents".

2.3.1 Matching architectures to desired agentic patterns

It is not necessary to have the agentic patterns and concepts of "'user feedback processing", "shared states", "independent "concurrent agents", and "error resistance" in the provided architectures, however, because they are desired within the agentic system, the architectures will be categorized by their ability to adhere to these agentic patterns.

A research survey by Acharya et al. (2025) provided three key architectural approaches that "autonomous agents use to accomplish their aims", they are: multi-agent systems, hierarchical reinforcement learning, and goal-oriented modular architectures. These three architectures will be considered for the creation of the different setups.

- A Multi-Agent system (MAS) consists of agents that work together to reach a desired output. A recent review of MAS by Amirkhani and Barshooi (2022) extended the definition of working together by elaborating that "the agents have to communicate, cooperate and coordinate with one another". The research draws the comparison that a multi-agent system works similarly to people in society. However, a recent survey by Plaat et al. (2025) stated that many multi-agent systems "assume a flat power hierarchy," which is not representative of a society.

- Hierarchical Reinforcement Learning (HRL) distinguishes between higher-level agents and lower-level agents. The higher-level agents (similar to managers) break up the task into smaller goals, which the lower-level agents execute. A research review by Hutsebaut-Buysse et al. (2022) adds that HRL is prone to have "behavior reuse" because the manager agent can ask a lower-level agent to redo the task.

- A goal-oriented modular architecture has one centralized planner agent that selects modules based on the answer planning. Each module (which may have an LLM) has a clearly defined role, however, the assignment of the task is given by the centralized planner agent, which ensures the main goal is met. The modules do not have to be called, can be rerun, and can be swapped out based on the requirements that the planner agent devised to reach the main goal. A survey about using a goal-oriented modular implementation for 6G technology by Getu et al. (2024) emphasized that the strength is gathering semantic information at the time that it is needed in such a way to achieve maximum efficiency.

The agentic concepts of "user feedback processing" and "error resilience" are determined on a framework level. However, the agentic patterns of "independent concurrent agents" and a "shared state" are (mostly) determined by the architecture.

A true multi-agent system has "independent concurrent agents" whereby the agents try to reach their individual goals with independent communication. HRL has higher-level and lower-level agents, so there needs to be some sequential ordering because the higher-level agent is executed first. However, it is possible to have concurrency between same-level agents. Goal-oriented modular architecture has modules that are often executed sequentially, for example, a planner, worker, and

solver module. An example is the ReWOO framework (Xu et al., 2023) because mixing the order of these can lead to unwanted outputs.

Often, a "shared state" is missing in goal-based modular architectures because rather than one shared state where all agents' outputs are gathered, the planner agent instructs individual modules. The multi-agent system architecture and HRL commonly have shared states. However, an HRL may exhibit the same behaviour as a goal-based modular architecture when the lower-level agents do not share a state with higher-level agents. Same-level agents may share a state.

2.3.2 Matching frameworks to desired agentic patterns and concepts

There are a large number of frameworks to choose from when designing setups. A framework defines how agents and tools are created and how they interact. In this background section, five suitable frameworks for this experiment are able to at least implement one of the above-mentioned architectures, it may have none of the agentic patterns or concepts. Often, the frameworks can create setups in multiple architectures depending on the methodological choices that are made. The frameworks are covered one by one, and it is explained which of the three architectures and four agentic patterns and concepts it can adhere to.

CrewAI[11] is a widely implemented framework within scientific contexts. For example, it was covered in two framework-comparison surveys from 2025 (Barbarroxa et al., 2025; Joshi, 2025). There are two types of CrewAI frameworks: a CrewAI Crew and a CrewAI Flow. Firstly, a general overview will be provided, and then the CrewAI Crew and CrewAI Flow will be explained in more detail separately.

A general overview of the differences can be found in Figure 3 from the GitHub documentation[12]. In the figure, it can be seen that the Crew is an environment where agents all have an individual LLM with a shared memory and tools. The agents accomplish separate tasks, and the outputs are combined to form the outcome. The Flow has a controlled order of execution, and it can execute entire Crews like agents. The CrewAI Crew has an automatically shared memory, no additional programming is needed, and the agents will orchestrate what of the memory they will use. The CrewAI Flow, however, all of the inputs and outputs of the agents need to be saved. The Flow is able to share a State, but this is limited to programmed variables. It is up to the programmer who is designing the Flow, to orchestrate exactly what is passed onto the agents or Crews in the Flow. This makes it more manual and restricted.

The Crew is a simple-to-use agentic environment where agent collaboration is a central element[13], it can adhere to all architectures and all agentic patterns and concepts. The Crew is able to have an HRL architecture (if a higher-level agent is introduced), a multi-agent system architecture, and a goal-oriented modular architecture. The Crew has the ability to automatically share states with a memory, and it has a system in place for automatic error handling. If there is an issue within an agent, it can be rerun or left out of consideration while the other agents continue to run. It is possible to run the agents concurrently or sequentially, and it has a built-in feedback module that can be used.

The Flow is created for "structured, event-driven workflows"[14]. The Flow is able to fulfil the

[11] https://www.Crewai.com/
[12] https://github.com/crewAIInc/crewAI
[13] https://docs.Crewai.com/concepts/Crews
[14] https://docs.Crewai.com/concepts/flows

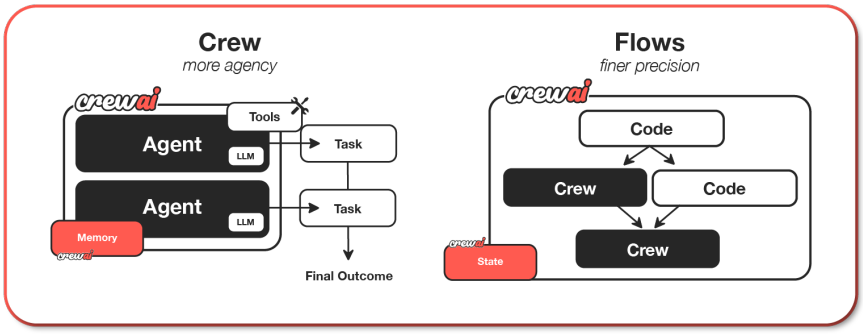

Figure 3: Comparison CrewAI Crew and Flow

criteria for a HRL and a goal-oriented modular architecture, but none of the four agentic patterns and concepts. The Flow is able to have a HRL architecture (if a higher-level agent is introduced) and a goal-oriented modular architecture. Inputs and outputs need to be programmed per agent, there is no shared memory, and the order of the agents needs to be pre-defined. These are important traits of a multi-agent system in terms of the "independent concurrent agents" and "shared state" agentic patterns. It is possible to add these agentic patterns onto the framework to become a multi-agent system, however, this may require extensive work. The CrewAI Flow is able to contain agents or entire Crews. User feedback processing is not built-in, but an agent can be added that can re-execute agents. The Flow does not have automatic error handling.

Similar to the CrewAI Flow is the LangChain[15] framework. LangChain is widely used, both within industry and scientific research, and it has much documentation. A few advantages are that it has "a rich ecosystem of tools"[16], ready-made agents, and integration packages such as langchain-openai. The architecture and agentic pattern possibilities are the same as those of the CrewAI Flow, for the same reasoning.

ReWOO and SmolAgents are both ReAct-based frameworks that first plan, then act, and then observe. ReWOO traditionally uses a planner agent, a worker agent, and a solver agent in sequential order, whereas SmolAgents uses a planning, acting, and solver module within each agent. First, ReWOO will be covered, followed by SmolAgents.

A ReAct-based framework called ReWOO (Reasoning WithOut Observation) was introduced by Xu et al. (2023) in 2023. It supports the goal-oriented modular architecture very well. In the original paper, the setup was described as a "modular paradigm". ReWOO supports a goal-oriented modular architecture because interacting agents (modules with LLMs) have a pre-defined execution loop[17]. The planner agent does all the thinking and devising of the steps, while the worker agent follows these steps, and the solver agent combines the results in this sequential order. ReWOO is not designed for HRL, because there are no direct sub-agents or manager agents. But, it is possible to create a planner with sub-planners if desired. This gives it a hierarchy. ReWOO is also not designed to be a multi-agent system because it "overlook[s] concurrency and shared states" (Malfa et al., 2025). These agentic patterns are important characteristics of multi-agent systems. The ReWOO framework does not have automatic error handling or user feedback processing.

[15]https://github.com/langchain-ai/langchain
[16]python.langchain.com/docs/introduction/
[17]https://github.com/billxbf/ReWOO

Huggingface Smolagents[18] framework was introduced in 2024 and will be considered as a potential framework since it is both open-source and compared in recent surveys of collaborative AI agent frameworks (Joshi, 2025; Satyadhar, 2025). Its strength lies in AI research and the simplicity of implementation. It is similar to CrewAI because it is able to support a multi-agent system architecture, a goal-oriented modular architecture, and it supports two of the four agentic patterns. Contrary to ReWOO, SmolAgents does not have to have a pre-defined agent planning. Rather, it has individual agents that can plan, act, and observe for themselves. These traits make it suitable for multi-agent systems and goal-oriented modular architectures. Smolagents has the ability to share states automatically with a memory and create independent, concurrent agents. It does not possess automatic error handling or automatic feedback implementation, however, this can be implemented by try/except statements (for error handling) and adding agents (for feedback implementation).

2.4 LLM model integration

To ensure correct functionality, the agents themselves and the available tools may make use of LLM calls. Two integration possibilities are covered below. Firstly, the LLM model integration for tools will be covered, and then the LLM model integration for agents.

As mentioned in background section 2.2, tools are "predefined code paths". The framework only provides an influence on the inputs and outputs. This means that the LLM calls in tools are predefined. For example, GPT_researcher uses GPT-4o-mini as its default model. More details can be found in the documentation[19].

In terms of LLM-integration of agents, the frameworks that were considered for the setups are all capable of integrating with a wide range of models, such as OpenAI, META-LLAMA, and Gemini. The choice of model may depend on choices made by previous research or the competitive advantage that one LLM model might offer.

3 Related work

In terms of using agent specialization to improve the internal and external domain knowledge gathering ability of agents, there is currently much research. Below, two research examples on the creation of a domain knowledge gathering agent in process mining are given. Followed by two research studies that create DKGA specialization in the domains of teaching assistants and process mining. Finally, there is work that also tests DKGA specialization to improve the output of cross-domain questions, but with a very different specialization technique.

In the process mining domain, external domain knowledge has been used to answer questions about an abstraction of an event log (which is a form of internal domain knowledge). A research by Fontenla-Seco et al. (2023) created a single DKGA called C-4PM, which used a ChatGPT call to answer questions about the abstraction of a process model. Two weaknesses of the DKGA were that the output did not have references and that the research focused on answering simple abstraction questions. These questions were able to be answered by one call, such as "Are there cases where [event x] happens after [event y]?", "explain the model of the process", and "does the

[18] https://github.com/huggingface/smolagents
[19] https://docs.GPTr.dev/docs/GPT-researcher/llms

model allow for any behavior?". It is stated that more extensive experiments have not been done and that they are potential for future work.

Another example of one DKGA that gathers external domain knowledge and combines it with internal domain knowledge is a research by Vogt et al. (2024). It implements the search tool GPT_researcher (see Background section 2.2.3) using a REWOO framework (see Background section 2.3) to provide external domain knowledge for process mining. The experiment is somewhat limited due to its lack of proper evaluation (the use of simple ablation testing and a subjective qualitative analysis) and because the agentic framework makes use of just one DKGA to answer very complex questions.

One work by Gong et al. (2024) investigated the ability of DKGAs *specialization* to gather knowledge across different domains. The result was that "specialized approach to domain-specific knowledge" improved the outputs significantly. However, the implemented strategy for specialization was very different. Instead of utilizing role prompting, the implemented strategy added "domain-specific modules" to the framework, which gave the models segmented *training data*.

A collaborative multi-agent research from 2024 created specialized DKGAs through role prompting to answer high school science questions (Chen et al., 2024). This specialization with different roles was an "effective strategy". For example, physics questions were answered with the sequential execution of three specialized agents: a physicist, a mathematician, and a summarizer. Since the physicist and the mathematician gathered knowledge, they can be identified as DKGAs. The summarizer used knowledge from these agents and, therefore, is not identified as a DKGA. One limitation of this research is the "task-specific design" which meant all the prompts were written by humans and rigid for a particular task. This will also be a limitation of this research since rigid role-prompting will be used so the DKGAs stay on task.

There is research by Berti et al. (2024) in the area of process mining that uses multiple DKGAs to gather external domain knowledge to perform a root cause analysis on a DFG abstraction. The researchers used a CrewAI Crew to make an "AI-Based Agents Workflow" which was a "powerful tool". The agents were a cause analyst, a cause explainer, and an insights grader (which gave a score to the process mining insights proposed by others). A downside of this research is that the agents in the workflow did not possess domain knowledge searching tools, meaning that the analysis did not reference the reasoning for the decision.

4 Method

Five setups are created that all answer the same process risk and process inefficiency questions. Three of these setups have the number of DKGAs as an independent variable, and the other two setups are part of a human-in-the-loop experiment by allowing the user to provide feedback.

A global setup can be seen in Figure 4, which holds for all five experimental setups. It has two inputs (an event log and a question), a total of N DKGAs which run sequentially, and a writing agent. This writing agent combines the domain knowledge gathered by all the DKGAs and weaves it together into a coherent report. This writing agent is the same for every setup with exactly the same prompting; its aim is to answer the process risk and inefficiency questions. The code of all the setups can be found on GitHub.

This section has five subsections that explain the methodological choices made to create the setups. The first subsection explains how the question input is created. To create DKGAs, three

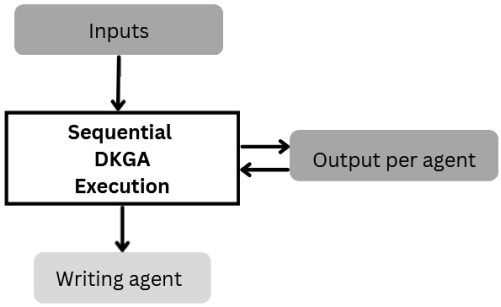

Figure 4: Global setup

elements are required: a clear role definition, a task, and available tools. The second subsection focuses on the role definition and task creation depending on the number of agents in the setup. Available tools are covered in the third subsection. The fourth subsection covers the architecture and framework selection made to create a desired environment for the DKGAs. The evaluation method is covered in the final subsection.

4.1 Process mining question design

The questions used in this particular research focus on a causal analysis of risks or inefficiencies for certain processes. Combining different areas of domain knowledge can help to spot these causes, as mentioned in the computer factory example in Background section 2.1, whereby the wrapping stage of computers is slow, and added domain knowledge may add that this is because computers consist of many fragile parts and that wrapping if often done by hand.

The experimental setups are all given the same six process risk and process inefficiency questions. All questions are made from the following question template:
"Can you find the **{focus}** in the **{process}** process at **{company}** and what are the potential causes for the **{focus}**?"

The variables that can be changed are inside the curly braces. This question template will test six variations of process risk and process inefficiency questions, namely:

- "Can you find the **environmental risks** in the **purchase to pay (P2P)** process at **Walmart** and what are the potential causes for the environmental risks?"
- "Can you find the **regulatory risks** in the **loan application** process at **Wells Fargo bank** and what are the potential causes for the regulatory risks?"
- "Can you find the **IT and cyber risks** in the **IT incident** process at **Volvo** and what are the potential causes for the It and cyber risks?"
- "Can you find the **audit risks** in the **order to cash (O2C)** process at **Procter & Gamble (P&G)** and what are the potential causes for the environmental risks?"

- "Can you find the **operational risks** in the **travel expenses** process at **Google** and what are the potential causes for the environmental risks?"
- "Can you find the **process inefficiencies** in the **accounts payable (AP)** process at **General Electric (GE)** and what are the potential causes for the environmental risks?"

In this paper, when referring to process risk or inefficiency questions that are answered, these are the exact formulations of those questions.

4.2 Creating DKGAs: role and task defintion

To be able to answer the six different process risk and process inefficiency questions, different areas of domain knowledge expertise are required. The role and task definition changes when more DKGAs are introduced.

When taking the example question "can you find the **environmental risks** in the **purchase to pay (P2P)** process at **Walmart** and what are the potential causes for the **environmental risks**?", it is possible to answer with one DKGA, two DKGAs, three DKGAs, and so forth by making use of role prompting and task creation to specialize agents. If there is only one DKGA that has to gather knowledge from the domains of "environmental risks", the "purchase to pay (P2P)" process, and the "Walmart" business, the agent is not specialized. The role would be a general expert on process mining and the task would be to gather knowledge about many different topics to be able to answer the questions. However, if more DKGAs work together, they can all gather a part of the knowledge and specialize in that domain. A theoretical example for the role specialization and task prompting of four DKGAs is the following:

1. Company expert: specialize in Walmart, taking the P2P process and environmental risks into account
2. Risks expert: specialize in environmental risks and its causes, taking Walmart into consideration
3. Process expert: specialize in the Purchase to Pay process, taking environmental risks into consideration
4. Process mining expert: take the event log of the P2P process and make an abstraction. Then report the most important events for environmental risks in the P2P process at Walmart.

The role specialization is the name of the expert mentioned before the colon, and the task is the sentence after the role specialization. In this theoretical example, the role specialization and task are less thorough than in the real experimental setups.

A metric of how specialized the agents are is the Degree of Specialization (DoS). The DoS is the number of agents that work together to gather domain knowledge minus one, because with only one DKGA, there is no specialization. In the example shown above, four DKGAs are given, so the degree of specialization is three.

4.3 Creating DKGAs: tool selection

For the creation of DKGAs, a tool selection is needed. Due to the reasoning in background section 2.2.3 about the potential overuse of tools, the DKGAs in this research will have a maximum number of two tools. This way, the agents have a clear oversight.

For all setups except one, the selected search tool for the DKGAs is the Tavily search tool due to its simple application and the fact that it is widely used within research. Tavily search returns the source, title, and a summary of the content of a page; an example can be seen in the Appendix section 9.4. The between-agent human-in-the-loop experiment will have a GPT_researcher as a search tool be able to measure the difference in reference gathering during the references analysis. GPT_researcher returns a report with references.

The selected process mining tool is the PM4PY library. All of the tools covered in the background have the necessary functionality, so this is based on ease of use and the fact that the PM4PY performed better in the mentioned comparison survey.

The DKGAs will be in control of when and how often their tools will be called to keep the independence of the agents in the multi-agent system. One exception is the between-agent human-in-the-loop experiment in which the DKGAs' use of tools is pre-defined.

4.4 DKGA environment: architecture and framework selection

Although hierarchical reinforcement learning and goal-oriented modular architectures are important in agentic AI, the focus of this research will be on multi-agent systems.

The reason a hierarchical reinforcement learning architecture is not chosen is because this research does not require a higher-level agent (such as a manager). Additionally, it could change the focus of the research, because a higher-level manager agent is in control of the inputs, outputs and what order to call the agents in. This means that the quality of the output is partially about how well the manager agent can handle different amounts of agents, not about the specialization increase of the DKGAs.

Goal-oriented modular architectures is not chosen for the main setups because the DKGAs need to be able to have individual goals and some level of autonomy. Modular architecture has a 'thinking' (planner) agent devises what tasks the modules will perform which restricts the freedom. For the between-agent human-in-the-loop experiment, this will be used for the reason that a multi-agent system cannot adhere to the strict agentic control it requires.

In a multi-agent system, the agents share a memory to "communicate and cooperate", but work towards performing their own goals, which they try to achieve as well as possible. In this research, a *sequential* multi-agent system is chosen for the rest of the setups, which misses the important characteristic of multi-agent systems. The "independent concurrent agents" agentic pattern of a multi-agent system is not met as well due to the sequential process, as mentioned in the Global Setup. The reason for a sequential process is to ensure comparability of outputs. For example, when running DoS2 multiple times in a non-sequential order, the outcomes may differ greatly. The agents are able to use the results of previous agents, if the previous agents differ, the output of certain agents is better than others.

In summary, a sequential multi-agent system is used for all setups but the human-in-the-loop between-agent setup because it is not possible to perform the desired action in a multi-agent system.

After a close examination of all the available frameworks for the experimental setups (see

background section 2.3.2), this research will be working with CrewAI. CrewAI is well-documented, creates a workflow that is very controllable.

For four of the five setups in this research, a CrewAI Crew will be used for two reasons: it adheres to all agentic patterns and concepts mentioned in the background section and there are multiple example research projects in process mining using CrewAI Crew, as was stated in the Related Work section 3.

The reason why the between-agent human-in-the-loop setup uses the Flow is because it requires specific agents to be re-run. This specific looping of agents is not possible for the CrewAI Crew which is more restrictive. A downside that follows is that there is no longer an automatically shared memory between the agents. In Section 2.3.2, it was explained that in the CrewAI Flow, there is no shared memory but a pre-programmed State, and that the inputs and outputs per agent also need to be programmed.

4.5 LLM model selection

The setups using the CrewAI Crew will make use of the default LLM, which is the GPT-4o-mini model, as can be found in the documentation[20].

The CrewAI Flow has no default model. The agent LLM integration needs to be explicitly written down in the code, such as "model=openai/GPT-4o". This gives more control for the model selection of each agent. The CrewAI Flow will also make use of GPT-4o-mini for all agents except the writing agent. The writing agent in the Flow will make use of GPT-4o because the strength of the final report depends on how well the writing agent can interlace all of the information. GPT-4o is a stronger model for text generation, so it is favoured over GPT-4o-mini.

As mentioned in the background information, tools are external pieces of code. They also have a hardcoded LLM call.

4.6 Evaluation

There are three chosen evaluation methods: the 'contribution percentage', the one-sided Wilcoxon signed rank test, and a references analysis. In a side study of this research, see Appendix section 9.1, three different evaluation methods were tested. This section briefly covers why some of the tested methods will not be used in this evaluation.

To analyse the quality of the experiment outputs, the quantitative 'contribution percentage' metric is set up. The specific definition is covered in Section 4.6.2.

Using the contribution percentage, a one-sided Wilcoxon signed rank test will be performed to compare whether the increase in DKGAs is statistically significant for the main experiment.

Another evaluation method that will be used is a references analysis, because it was a successful method in the side study.

An evaluation method that will not be used is the LLM-as-a-judge method because it was not yet reliable enough, as was shown in the side study. The LLM-as-a-judge would re-interpret the definition of %IDK (a measure similar to the 'contribution percentage') at every run, which led to large standard deviations. Also, the output was less explainable than human grading because the LLM did not provide the exact textual grading.

[20] https://docs.crewai.com/en/concepts/llms

Category	Description
Self-published company papers	This means that the source is from a page of the company that is getting analyzed. For example, www.media.volvocars.com for the Volvo prompt and www.pgsupplier.com for the P&G prompt
Academic publications	These references link to an acknowledged publications platform such as www.tandfonline.com and link.springer.com.
Non-academic research	References that lead to research done by companies. Often, these are large consulting companies that have done extensive market research. To qualify as "research," the paper needs to have links to other research or be done by multiple researchers. It cannot be, for example, a blog of one person who has done a small internet search. This is a fine line, yet this definition should really encapsulate research that has been done with significant effort, significant evidence or referencing, and/or a significant number of contributors.
Non-academic source	Any source that is non-academic and that does not fall in the category above falls into this category. Often, these are blogs of people who have done research into the definition or process in a particular domain.
Social media or press release	This includes articles and news publications that are published on social media platforms or press platforms. References from social media platforms are often from LinkedIn, and press release platforms range from papers such as "The Guardian" to "Forbes."
Link does not exist (anymore)	This is a link that does not lead to an article. It may be that the link once existed and that the company has removed the article, or it may be that it has never existed at all.
Law documents	This is a small category made for law documents. These are official documents that factually state the laws.

Table 1: Category explanations of references classification

Another method that will not be used is the percentage of integrated domain knowledge. This was an effective measure in the side study; however, because the main focus of this research differs, a new metric called the 'contribution percentage' is defined below, which will be applied. A tip from the side study that will be taken into account is to ensure that the measurement is well-defined. The less open to interpretation the definition is, the lower the chance of large standard deviations between measurements.

4.6.1 Reference categorization

The references categorization has a manual classifying process, and all the references are checked one by one. This means that even though the results are quantitative, several qualitative remarks can be made.

There are seven categories for the references, which have been made during the evaluation side study, and the table of categories with their explanations can be found in Table 1.

4.6.2 Contribution percentage

The contribution percentage is an evaluation metric that will be used to assess the quality of the textual output. Since the goal of the experiments is to answer process risk and process inefficiency questions with gathered domain knowledge, the metric measures this.

The contribution percentage is measured per paragraph with rules that are as follows:

1. Is the knowledge referenced to a process mining step?
2. Is the knowledge cited?
3. Is the information too generic or non-contributional?

Some extra explanation of these questions is:

1. The foundation of answering the question is looking at the process steps from the abstraction that are going to be analysed. The process steps are often represented with an arrow between two events, for example "Order Validation -> Customer Credit Check". These steps are taken from the process mining agent.
2. The DKGAs can use a search tool to gather information about their domain. In every paragraph, if domain knowledge is used, there should be a reference to the website or paper.
3. **Too generic** is non-cited information. An example of generic is "The Loan application process at Wells Fargo Bank is critical"[21] and "The management of travel expenses is a critical area for organizations, especially those operating on a global scale like Google"[22].
   Non-contributional means that the output contains information that is not relevant for answering the question or repetitive. For example, the *solution* for identified risks was never asked in any of the questions. However, it is covered in the sentence: "Further assessment and re-engineering of this process step can decrease the manual interventions often required, thereby reducing the operational bottlenecks associated with high rejection rates."[23]. Another non-contributional statement is when the content of the report is summarized. This is, for example "AP encompasses crucial activities such as invoice processing [...] and exception handling, all of which are intertwined with operational efficiency and vendor relationships"[24]. It does not add any new knowledge from the agents, and because the report is only a couple of pages, it does not need a summary of the content.

Below are three example calculations of the 'contribution percentage'. The calculation is simply taking a percentage of the characters that adhere to the definition, divided by the total number of characters. The characters in the curly braces adhere to the definition.

- {##1. Order Entry -> Order Fulfilment The first step in the O2C process at P&G is **Order Entry**, where customer orders are received and swiftly recorded. Audit risks in this phase are often associated with inaccuracies in data entry. Errors here can lead to prolonged delays

[21] DoS2_DK/DoS2_run2/regulatory-loan-wellsfargo
[22] DoS2_DK/DoS2_run1/operational-Travel-Google.txt
[23] DoS4_DK/audit-O2C
[24] DoS2_DK/DoS2_run1/inefficiencies-AP-GE.txt

in subsequent steps such as **Order Fulfillment**. According to HighRadius (2024), timely and accurate order processing is essential to avoid bottlenecks that can result in missed delivery dates, which directly correlates with the reputation of P&G. An error in an order may lead to customer dissatisfaction and necessitate corrective actions (Emerald Insight, 2022).} Implementing double-checking mechanisms or automated data entry systems can mitigate these risks effectively.[25]
Analysis: the first part of the paragraph covers a chosen step from the process mining agent, then continues with information from the process agent and finishes with a company repercussion. All the information has a reference. The last part of the paragraph is a recommendation for mitigations, this is not part of the event log question so this will not be covered in the coverage percentage. The percentage is $(646/756) * 100 = 85.5\%$.

- {Walmart's Purchase-to-Pay process reveals numerous environmental risks stemming from its operational and supply chain dynamics.} Notably, through rigorous sustainability initiatives and best practices, Walmart has viable avenues to reduce these risks significantly. Through careful management of requisitioning, vendor selection, order creation, receipt of goods, payment processing, and supplier management, Walmart can strengthen its commitment to environmental sustainability while improving operational efficiency. {The results derived from this analysis underline the necessity to incorporate sustainability as a fundamental aspect of business strategies in the P2P process.}[26]
  Analysis: This conclusion covers the answer to the event log question in the first sentence. The following phrase is becoming too general because the event log question did not ask for recommendations. The final sentence does contribute to the event log question by answering it. The contribution percentage is $(287/677) * 100 = 42\%$

- In the rapidly evolving business landscape of 2025, Accounts Payable (AP) has transcended its traditional role as a back-office function to become a strategic lever for cost control, risk management, and operational efficiency. For multinational corporations like General Electric (GE), optimizing AP processes is crucial to maintaining competitive advantage. This report synthesizes insights from process mining, industry benchmarks, and case studies to provide a comprehensive analysis of AP inefficiencies at GE and actionable strategies for improvement. The report draws on over 20 authoritative sources and integrates findings from process mining diagnostics to highlight key areas for optimization.[27]
  Analysis: The first sentence is too generic and does not have a reference for this statement. Following this is a sentence that introduces the topic of GE with AP processes. This sentence is non-contributional because it is a global statement that could be said about any process; there is no reference to back this up. However, it may be argued that it does contribute to answering the question by linking GE to the AP process. The example concludes with a non-contributional summary and a too generic explanation of the steps taken. The contribution percentage is $(0/704) * 100 = 0\%$ when examining the paragraph critically.

[25] DoS2_DK/DoS2_run2/audit-O2C-PG
[26] DoS0_DK/environment-P2P-Walmart
[27] DoS2_DK/DoS2_HITL_flow/Flow-AP-GE.txt

alpha values							
n	0.001	0.005	0.01	0.025	0.05	0.10	0.20
5	--	--	--	--	--	0	2
6	--	--	--	--	0	2	3
7	--	--	--	0	2	3	5
8	--	--	0	2	3	5	8
9	--	0	1	3	5	8	10
10	--	1	3	5	8	10	14

Figure 5: Critical values table for N=6

4.6.3 Wilcoxon signed rank test

A paired one-sided Wilcoxon signed rank test has been selected to compare the benchmark setup (DoS0) to the DoS2 and DoS4 setups. The reason for the signed Wilcoxon test is because the variables are dependent[28] and because of the small amount of samples, no assumption can be made about the underlying distribution. The reason a t-test was not chosen is because it assumes that paired differences of the questions should be normally distributed. It also and does not handle outliers well.

The tests performed will be one-sided because the test is only interested in whether the increase of DKGAs **improves** the contribution percentage. If it improves the contribution percentage, the median of the differences is negative.

The selected significance level for the H_0 to be rejected is $\alpha < 0.05$. The reason for this strict significance level is that the definition of contribution percentage can be quite subjective. This subjectivity can lead to large standard deviations per grading attempt, thus, it is important to be sure of the hypothesis rejection.

The calculation that will be used is as follows: calculate the contribution percentage difference, assign a rank based on sign and size, calculate W and compare it to the critical value for the desired significance level. These steps will be described in more detail in the following paragraphs.

The contribution percentage difference is calculated with the formula $d_i = DoS2[i] - DoS0[i]$.

Using this, ranks are assigned to the difference in contribution percentage based on the size of the difference and its sign. Out of n pairs, the smallest difference is assigned a rank of 1, and the largest difference is assigned a rank of n. The sign is added after the rank is determined. For example, if the differences are -3, -4, 8, 10, -12, and -13, the ranks are: -1,-2,3,4,-5,-6. The Wilcoxon test needs non-zero differences to rank, so the test cannot be performed if there are many differences of zero. If the differences are the same, those values receive the same rank.

The W calculation is finding the minimum of the sum of the ranks. See the formula below.

$$W = \min\left(\sum_{d_i>0}^{i} \text{positive\_rank}(|d_i|), \quad \sum_{d_i<0}^{i} \text{negative\_rank}(|d_i|)\right).$$

The critical value can be found by looking at Table 5, the significance level can be matched with the N. However, because this Wilcoxon signed rank test makes use of a one-sided experiment, the chosen significance level is doubled. Since the experiments in this research work with six pairwise

[28]The DoS0 and DoS2 contribution percentages gathered are answering the same event log question, they cannot be seen as independent from each other

differences (six evaluated questions), N=6, and, as mentioned beforehand, the chosen significance level was 0.05. Doubling this gets a critical value for the experiments is 2.

If W is below or equal to two, the outcome is significant and the null hypothesis is rejected. Otherwise, it is not significant, and the null hypothesis is not rejected.

The hypotheses are as follows:

- H_{0_DoS2} :There is no difference in performance between DoS0 and DoS2 (median difference = 0).
- H_{1_DoS2} : DoS2 outperforms DoS0 (median difference > 0).
- H_{0_DoS4}: There is no difference in performance between DoS0 and DoS4 (median difference = 0).
- H_{1_DoS4} : DoS4 outperforms DoS0 (median difference > 0).

5 Experimental setup

The objective of this research is to evaluate how DKGA specialization through specific role-prompting and user feedback processing affects the quality of reports for answering process risk and process inefficiency questions.

By varying the amount and roles of DKGAs, the main experiment changes the degree of specialization per agent. Three setups are tested:

- DoS0: A single generalist agent
- DoS2: Three moderately specialized agents
- DoS4: Five highly specialized agents

The setup with the degree of specialization of zero is a simplified version of the setup in previous research by Vogt et al. (2024).

Two DoS2 setups are specialized further by the implementation of different human-in-the-loop feedback methods to test whether this improved the output quality. These setups can be described by the following:

- DoS2CH: DoS2 setup with within-agent feedback
- DoS2FH: DoS2 setup with between-agent feedback

The within-agent feedback works as follows: after a DKGA has run, the user is presented with the result of the DKGA and is allowed to provide feedback. The DKGA implements the feedback. This repeats until the user has no more feedback.

The between-agent setup first runs through all agents and creates a final report. The user can read the report and formulate feedback based on it. It can ask the setup to rerun a particular agent with supporting feedback. Once the particular agent is re-run, the setup will return to the writing agent and execute it, so the new information gets recombined. This repeats until the user has no more feedback.

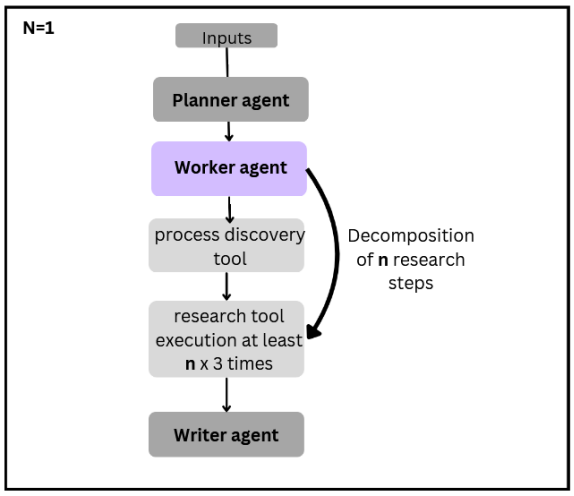

Figure 6: Degree of specialization is zero (N=1) including planner agent

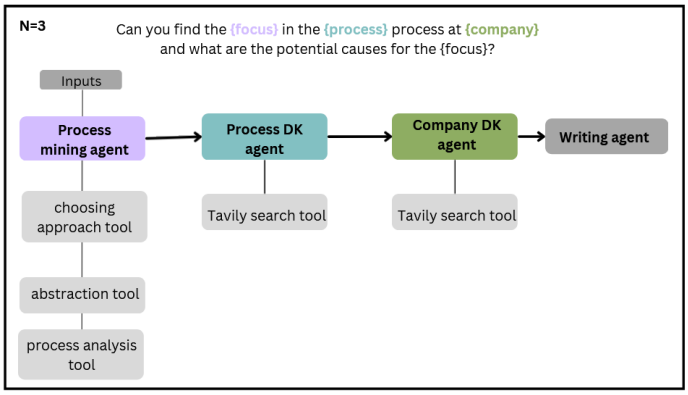

Figure 7: Degree of specialization is 2 (N=3)

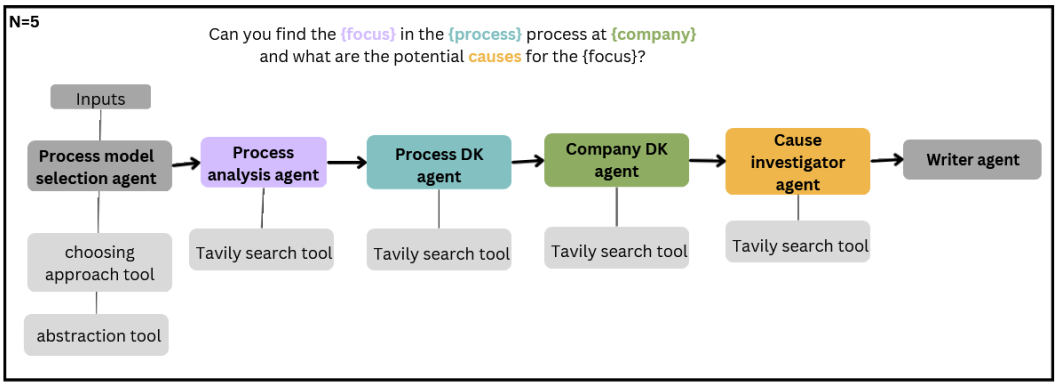

Figure 8: Degree of specialization is 4 (N=5)

5.1 Main experiment setup

This experiment changes the Degree of Specialization (DoS) of DKGAs through role-prompting. The setups are as follows:

- Degree of Specialization 0 (DoS0): one worker agent without a specialization. This is the baseline.
- Degree of Specialization 2 (DoS2): one process mining agent, one process agent, and one company agent.
- Degree of Specialization 4 (DoS4): one process model selection agent, one process analysis agent, one process specialist agent, one company agent, and one cause investigator agent.

A visual abstraction of the three setups can be seen in Figure 6, 7 and 8.

5.2 Human-In-The-Loop (HITL) experiment setup

The HITL experiment provided the DKGAs with the human trait of interaction by allowing the user to provide feedback. As mentioned in the method (section 4.4), the two setups have a different architecture, framework, and reference-gathering tool.

The DoS2CH within-agent feedback setup has the same DKGAs and search tools as in the conceptual framework of DoS2. The DoS2FH between-agent feedback setup uses a different architecture, framework, reference-gathering tool, and number of agents than the within-agent setup. Firstly, the DoS2CH within-agent setup will be explained in more detail, and then the DoS2FH between-agent setup.

For the DoS2CH setup, the framework makes use of a CrewAI Crew framework with a sequential multi-agent system architecture (4.4). To gather external domain knowledge, the Tavily search tool is used, as explained in the tool selection subsection of the method 4.3. In the Crew, the human-in-the-loop implementation is as simple as adding "human_input: True" to the DKGAs in the task definition file.

An explanation for the selection of a goal-based modular architecture and a CrewAI Flow framework is provided in method section 4.4 and the explanation for the GPT_researcher tool can be found in method section 4.3.

The two extra agents within DoS2FH are: an introduction agent to gather the question information and a human agent to receive and process feedback. The reason for a separate introduction agent is that the CrewAI Flow setup reruns the entire agents. In other setups, the first DKGA gathers user information, and the feedback implementation of the Crew does not ask for the input again. The introduction agent will not have any searching tools to prevent the agent from performing abnormally.

The Flow does not have a shared state; instead, it shares a pre-programmed class instance. In this experimental setup, it is defined as:

```
class FlowState(BaseModel):
    process: str = "-"
    company: str = "-"
    focus: str = "-"
    chosen_approach: str = "-"
    filename: str = "-"
    specific_question: str = "-"
    human_feedback: str = "-"
    finished: bool = False
```

FlowState is the name of the class instance that is passed through all the agents.

The feedback is implemented as follows. The nested attribute called "finished" starts with the boolean *False* and gets changed to *True* once the full agent sequence has been run once. If the nested attribute "finished" is *True*, then the next agent in the sequence is always the writing agent. If it is *False*, the sequence continues.

FlowState contains more variables besides "finished". *Chosen_approach* is the selection the abstraction agent makes. Company, process, focus, and filename will be added by the user in the introduction agent, and the other variables are added by the agents when running. For example, human feedback is added by the human (feedback) agent.

Setup	Agent	Functionality	GPT LLM
DoS0	Planner	Main	4o-mini
DoS2FH	Introduction	Main	4.o
DoS0	Worker	Main	4o-mini
DoS2FH, DoS2, DoS2CH	Process Mining	Choosing approach tool	4o-mini
DoS4	Process model selection	Choosing approach tool	4o-mini
DoS2FH, DoS2, DoS2CH	Process Mining	Analysis tool	4o-mini
DoS4	Process analysis	Analyses abstraction	4o-mini
DoS2FH, DoS2, DoS4, DoS2CH	Company DK	Main	4o-mini
DoS2FH, DoS2, DoS4, DoS2CH	Process DK	Main	4o-mini
DoS4	Cause investigator	Cause agent	4o-mini
DoS0, DoS2, DoS4, DoS2CH	Writing	Main	4o-mini
DoS2FH	Writing	Main	4o
DoS2FH	Human	Main	4o-mini

Table 2: Overview of each LLM-based part of the setups

5.3 Creating the abstraction from an event log

In the DoS2, DoS2CH and DoS2FH setups, a process mining agent is present which is able to recognise .csv and .xes files as event log inputs. In DoS4, this is the process model selection agent. Using this input, the PM4PY library takes an abstraction and turns the event log into either a directly-follows graph, a temporal profile graph, or a variants graph depending on the context of the process risk and process inefficiencies questions. Figure 14, in the Appendix, shows an example DFG abstraction by the process mining agent.

After the abstraction has taken place, an analysis of this process is executed either by an independent agent (DoS4) or a tool (DoS2, DoS2CH,DoS2FH). An example output can be seen in Figure 15, in the Appendix.

5.4 LLM usage in all experiment setups

As mentioned in the method, agents in the CrewAI Crew use the default model GPT-4o-mini. There is no default in the CrewAI Flow, however, GPT-4o-mini has been chosen because it was the default of the CrewAI Crew. An exception for this is the writing agent, which makes use of GPT-4o (see methodology section 4.5).

In the CrewAI Flow and the tools of both frameworks, the LLM calls are hardcoded, such as model="openai/GPT-4o". The tools used in the Crew and the Flow setups have the same LLM calls.

An overview of the LLM use linked with the agent, and if applicable, the tool can be seen in Table 2.

5.5 Evaluation steps

All five experimental setups have at least six contribution percentage measurements because there are six process risk and process inefficiency questions. Two exceptions are the DoS0 and DoS2

Focus	Process	Company	Run 1	Run 2	Run 3	Average	Sd σ
Environmental risk	P2P	Walmart	54.67	60.90	57.19	57.18	3.13
Audit risk	O2C	P&G	56.65	61.55	61.94	60.05	2.95
Process inefficiencies	AP	GE	44.46	41.22	63.40	49.69	11.98
Operational risks	Travel cost	Google	64.13	66.92	51.66	60.90	8.13
Regulatory risk	Loan app.	WF bank	52.55	54.80	70.04	59.13	9.52
IT & Cyber risk	IT incident	Volvo	69.93	65.36	55.42	63.57	7.42

Table 3: Contribution [%] for multiple DoS0 runs

setups, which are be run three times to be able to give a more reliable average confidence percentage per run. This decision is based on a recommendation from the side study (Appendix section 9.1), which stated that "it is recommended to use one human evaluator who repeats the evaluation multiple times". Running the setup three times means that all six questions are asked three times.

DoS2CH and DoS2FH will have a reference analysis, and a one-sided Wilcoxon signed rank test will be performed on the results from the main experiment.

6 Results

This chapter covers the results of the main experiment and the human-in-the-loop experiment. Both experimental results have a quantitative and qualitative description.

The main experiment has an additional Wilcoxon signed rank test, and the human-in-the-loop experiment is subjected to a references analysis.

The final subsection, section 6.3, contains a summary of the results.

6.1 Main experiment

In the main experiment, the DoS0 and DoS2 setups were each run three times, as explained in the Experimental Setup Section 5.1. The corresponding percentages can be found in Tables 3 and 4, along with the average and standard deviation for each question. The grey box in Table 4 indicates that this particular experiment was rerun due to the contribution percentage being less than 10% in total. This was abnormal and did not happen on any of the other runs.

Table 6, shows the (average) contribution percentages of the DoS0, DoS2 and DoS4 setups. The colours in the table represent the differences in percentage to the DoS0 results, the explanation can be found in the Table 5. DoS0 is considered to be the baseline so the colour representations of DoS2 and DoS4 setups are compared to DoS0.

6.1.1 Quantitative results description

Four main quantitative descriptions will be covered: the differences in average contribution percentage between the setups, describing specific differences between the setups, finding the extreme values in Table 6, and a description of the standard deviations between runs of the DoS0 and DoS2 setups from Table 3 and 4.

Focus	Process	Company	Run 1	Run 2	Run 3	Average	Sd σ
Environmental risk	P2P	Walmart	85.67	82.04	76.97	81.56	3.57
Audit risk	O2C	P&G	88.73	84.74	78.12	83.86	4.38
Process inefficiencies	AP	GE	69.43	67.33	68.95	68.57	0.90
Operational risks	Travel cost	Google	83.47	57.54	71.19	70.73	10.59
Regulatory risk	Loan app.	WF bank	88.13	93.23	85.84	89.07	3.09
IT & Cyber risk	IT incident	Volvo	70.12	72.27	67.18	69.86	2.09

Table 4: Contribution [%] for multiple DoS2 runs

Colour	Difference in percentage	Explanation
	$>= 10\%$	Compared to DoS0, the CP is at least 10% higher.
	$0\% <= x < 10\%$	Compared to the DoS0, the CP is between 0 and 10% higher
	$< 0\%$	Compared to the DoS0, the CP is less than 5% higher, which can also mean that it is negative compared to DoS0.

Table 5: Colour definition of Contribution Percentage (CP) comparison tables

Table 6 shows that the contribution percentage average for the DoS2 setup is 18.86% higher than that of DoS0, while the difference between the DoS0 and the DoS4 setup is only 11.66%. The percentage contribution rises between the DoS0 and DoS2 setup and then drops between the DoS2 and the DoS4 setup.

When examining Table 6 in more detail, compared to DoS0, four of the DoS2 questions had at least 10% higher contribution percentage, and the other two had a difference between five and ten percent higher. The lowest DoS2 percentage is 69.86% of the IT incident handling process with the IT & cyber risks and Volvo as the company.

Compared to DoS0, DoS4 had four instances that performed better in terms of contribution percentage. Three instances have a minimum of 10% higher and one instance is between five and ten percent higher. Two instances had a percentage difference of -13.6% and -1.03% compared to DoS0. The lowest DoS4 percentage is the lowest of the whole table as was mentioned before. Compared to DoS2, DoS4 has only lower percentages of contribution.

In terms of the highest and lowest values of Table 6, the highest value of the table is from the DoS2 setup with the regulatory risk focus; the contribution percentage is 89.07%. The lowest value of the table is 36.09% from the DoS4 setup, which examines the process inefficiencies of the AP process for General Electric. This is the lowest value of the table by 13.6%, thus could be considered as an outlier compared to the rest of the data. But, it could also be part of a trend because all of the lowest values of DoS0, DoS2, and DoS4 are all answering the AP process question.

Table 3 has a large standard deviation of 11.89% for the question with the focus on process inefficiencies. The table only has two standard deviations below five percent. Table 4, which presents the results from the multiple DoS2 runs, shows lower standard deviations. There is only one standard deviation above five percent. The highest standard deviation in the table is 10.59%.

Focus	Process	Company	DoS0	DoS2	DoS4
Environmental risk	Purchase-to-Pay (P2P)	Walmart	57.18	81.56	78.83
Audit risk	Order-to-Cash (O2C)	Procter & Gamble (P&G)	60.05	83.86	67.51
Process inefficiencies	Accounts Payable (AP)	General Electric (GE)	49.69	68.57	36.09
Operational risks	Travel expenses	Google	60.90	70.73	59.87
Regulatory risk	Loan application	Wells Fargo bank	59.13	89.07	81.45
IT & Cyber risk	IT incident handling	Volvo	63.57	69.86	69.96
Average of DoS			58.42	77.28	65.62

Table 6: Contribution [%]; a table for comparing DoS0, DoS2 and DoS4

6.1.2 Qualitative results description

In this subsection, three main topics will be covered: a qualitative description of the highest and lowest values, an output analysis of why 100% contribution percentage cannot be reached and two functionality descriptions.

The highest value in the table is from the DoS2 Loan application process. The output paper is exemplary in contribution because almost every paragraph has a cited reference to the company, the process, and the step in the event log. The two outputs that scored the lowest contribution percentage had the issue that company-specific information gathered by the company DKGA was left out in some paragraphs. The results of this DKGA were overlooked in the writing agent when it combined all the DKGA results.

There are three reasons why all contribution percentages are lower than 90%: the introductions and conclusions being too general, the summaries and an incorrect interpretation of the"PM_agent".

The introductions and conclusions in the outputs are often too general. Statements such as "AP encompasses crucial activities such as [...], all of which are intertwined with operational efficiency and vendor relationships" or "For Wells Fargo, a leading bank in the United States, understanding bottlenecks in this process can significantly enhance customer satisfaction and improve operational efficiency" are examples of this.

If the introduction summarizes the content of the report, it is not counted as a contribution, as was mentioned in the definition. In some outputs, this led to a lower contribution percentage than neccesary.

Sometimes, the writer agent interpreted the name of the process mining agent, the "PM_agent", as a project management agent. It added the incorrect name of the agent to its own prompting and this incorrect assumption was another reason that some scores were lower.

Two other important aspect to note in terms of functionality are the incorrect length of the output and the correct Tavily execution. The reports are no longer than 1500 words, even though the prompt instructs to have more than 1500 words. The Tavily search tool used by the DKGAs correctly returned 20 references, as instructed by the tasks. An example can be seen in the Appendix Section 9.4.

Focus (i)	$\lvert DoS2[i] - DoS0[i] \rvert$	**Signed rank**	$\lvert DoS4[i] - DoS0[i] \rvert$	**Signed rank**
Environmental risk	24.38	+5	21.65	+5
Audit risk	23.81	+4	7.46	+3
Process inefficiencies	18.88	+3	13.60	-4
Operational risks	9.83	+2	1.03	-1
Regulatory risk	29.94	+6	22.32	+6
IT & Cyber risk	6.29	+1	6.39	+2

Table 7: Pairwise difference (d_i) in contribution percentage [%]

6.1.3 Wilcoxon signed rank test

As mentioned in the method section 4.6.3, the significance test will be one-sided due to the fact that the test is whether DoS2 and DoS4 perform *better* than DoS0. The python code used to calculate the p-value can be found in the evaluation repository[29].

The six process risk and process inefficiency questions will be evaluated by making pairs of the contribution percentages per setup, in the case of DoS2 and DoS0 these values are the averages of the three runs that are performed.

The outcome of the statistical test is shows that DoS2 significantly outperforms DoS0. The reasoning for this is as follows: all values of DoS2 are larger then DoS0 ($d_i = DoS2[i] - DoS0[i]$), and they are positive. The assigned ranks therefore are $+1, +2, +3, +4, +5$ and $+6$. The W value is zero because the minimum of the sum of the positive and the negative ranks is zero ($W = min(0, 21) = 0$). As shown in Table 5, the critical value for W at N=6 with a significance level of p=*0.05* is 2. W is below the significance level so the result is significant at p ¡ .05. The proposed null hypothesis - there is no difference in performance between DoS0 and DoS2 - is rejected.

DoS0 has three runs and DoS4 has one test run. The averages of the DoS0 runs will be compared with the results from DoS4, set of 6 pairwise differences. The outcome of the statistical test is that DoS4 does not significantly outperform DoS0. as The assigned ranks are $+2, +3, +5, +6$ and $-1, -4$. The sum of the absolute values of the ranks are 16 and 5. This makes the W value calculation:$W = min(16, 5) = 5$. As shown in Table 5, the critical value for W at N=6 with a significance level of p=*0.05* is 2 thus W is not below the significance level so the result is not significant. The null hypothesis - there is no difference in performance between DoS0 and DoS4 - is not rejected.

6.2 HITL experiment

There are two setups for the HITL experiment, DoS2CH and DoS2FH, and they will be compared to the DoS2 setup. The comparisons in contribution percentage can be found in Table 8 and the colour definitions of Table 5 are used. There are different results based on the setups. The Crew and Flow setups both have a report output [30][31]. The Flow has three additional outputs of:

[29] https://github.com/CodingBuddy25/Evaluation_LLM/blob/main/WilcoxonTest.py
[30] https://github.com/CodingBuddy25/CrewAI_DK4PM/tree/flow_DoS2_HITL/Results
[31] https://github.com/CodingBuddy25/CrewAI_DK4PM/tree/crew_DoS2/dinsdag/Results_storage

Focus	Process	Company	DoS2	DoS2FH	DoS2CH
Environmental risk	P2P	Walmart	81.56	65.69	82.02
Audit risk	O2C	P&G	83.86	62.28	89.34
Process inefficiencies	AP	GE	68.57	52.96	73.14
Operational risks	Travel	Google	70.73	70.24	78.52
Regulatory risk	Loan	Wells	89.07	52.65	87.73
IT & Cyber risk	IT	Volvo	69.86	56.09	77.41
Average of DoS			77.28	59.99	81.36

Table 8: Contribution [%] for the DoS2 setup compared to DoS2FH and DoS2CH

intermediate agent outputs[32], a rendered Flow diagram (see Appendix section 17) and an event log to keep track of when the agents are finished (in Appendix section 9.5.1).

6.2.1 Quantitative results description

As is clearly illustrated in Table 8, the DoS2FH setup has lower contribution percentages compared to DoS2. The average decrease is 17.29%. On the other hand, the HITL Crew is, besides one outlier in the Loan application process, improving the DoS2 output. However, not with an increase higher than 10%. The lowest improvement is 0.46%, which is negligible as an improvement. The highest increases in contribution percentage are 8.28% of the Travel costs process and 7.55% of the IT incident process. The average increase in contribution percentage is 4.08%.

6.2.2 Qualitative results description

In this subsection covers three main topics: a common problem in the written output, two non-quantitatively measured benefits and four feedback implementation descriptions.

All the DoS2FH outputs added a chapter about process mining as a tool. The agents were expected to use intermediate results from the process mining agent, rather than conducting research on the topic.

Two non-quantitatively measured benefits for the Crew and Flow setups are that they both increased the explainability and the transparency of the system output for the user.

The implementation of feedback rounds created a more user-friendly setup because the user has the ability to interact with the system and steer the output into the desired requirements by rerunning agents if necessary.

Additionally, the human-in-the-loop provided more transparency to the user. To be able to give feedback, the user is shown the intermediate DKGA outputs. From these outputs, it is clear where the information from the final report is gathered from.

Four feedback implementation descriptions are given below. Firstly, an unsuccessful feedback implementation of rerunning the reference gathering is covered. This is followed by a successful implementation of the process. Then, the benefits of specific feedback are given. Finally, the importance of removing of unwanted open-ended statements to avoid misdirection is covered.

[32]https://github.com/CodingBuddy25/CrewAI_DK4PM/tree/flow_DoS2_HITL/dinsdag/Intermediate_results

Despite the added user friendliness, the DoS2FH setup experienced difficulties implementing feedback. General feedback, such as "can you add more references related to IT incident handling?" often failed. The setup changed the naming of the references instead of gathering new references. An example of the incorrect feedback implementation is changing the following reference "Business process mining: An industrial application" to "Business process mining: An industrial application relating to IT incident management". The two main problems of this modification are that the reference no longer exists and that the content of the source is still the same, thus unrelated to IT incidents.

A successful references feedback implementations was a prompts such as "always provide the full source link". This is because the knowledge is already accessible from the Tavily search result, the DKGA just did not select it in their output. With this specification, the reference links are added correctly.

DoS2CH performed well when the feedback was specific. For example, the phrasing "replace reference x and use the Tavily search to do this" was well-implemented. It refers specifically to a place in the output and it makes sure the domain knowledge gathering tool is called to reduce the chance of hallucination.

Another effective feedback method was to catch errors of the DKGAs early. Occasionally, the results of the DKGA includes a sentence about possible process recommendations or improvements. The writer agent would sometimes interpret these sentences as "also provide process improvements", thus the report would have a section that does not contribute to the contribution percentage. These sentences can be removed using the simple feedback of "please remove sentence x". In all but one of the cases, this feedback strategy was useful. However, within the event log question about Volvo, an improvement section was added despite the removal of a recommendation sentence. This reduced the overall contribution percentage.

6.2.3 References analysis

For the DoS2 and DoS2FH setups, all references within the output files were manually classified. As was mentioned in the method, DoS2 makes use of a Tavily search tool, whereas DoS2FH uses GPT_researcher to gather information. An overview of differences in the categorization can be found in Figure 9. There are several things to note about this graph: the None count, the diversity of reference origin, and the references count.

As can be seen in Figure 9, the DoS2 setup has numerous references that link to non-existent webpages (None). For example, the None count for the Loan Application process question is nine out of 20 references and the count for the AP process is seven out of 21. In total 20 out of 92 references are in the None category. On the other hand, the DoS2FH setup in total two references that do not lead anywhere. One is from the IT_cyber process and the other from the order to cash process. Only two of the 95 references are classfied as None.

An unexpected result is that the DoS2FH setup collects from a slightly broader range of reference categories. The amount of reference categories collected from for the Flow in ascending order is $\{2, 3, 4, 5, 5, 6\}$ and for the Crew is $\{2, 2, 3, 4, 4, 5\}$.

Finally, there are very few references in total. Four of the six papers produced by the DoS2 setup have a total reference count is less than 15, which is not in line with what the writing prompt instructs. The DoS2FH setup has two results with a reference count lower than 15.

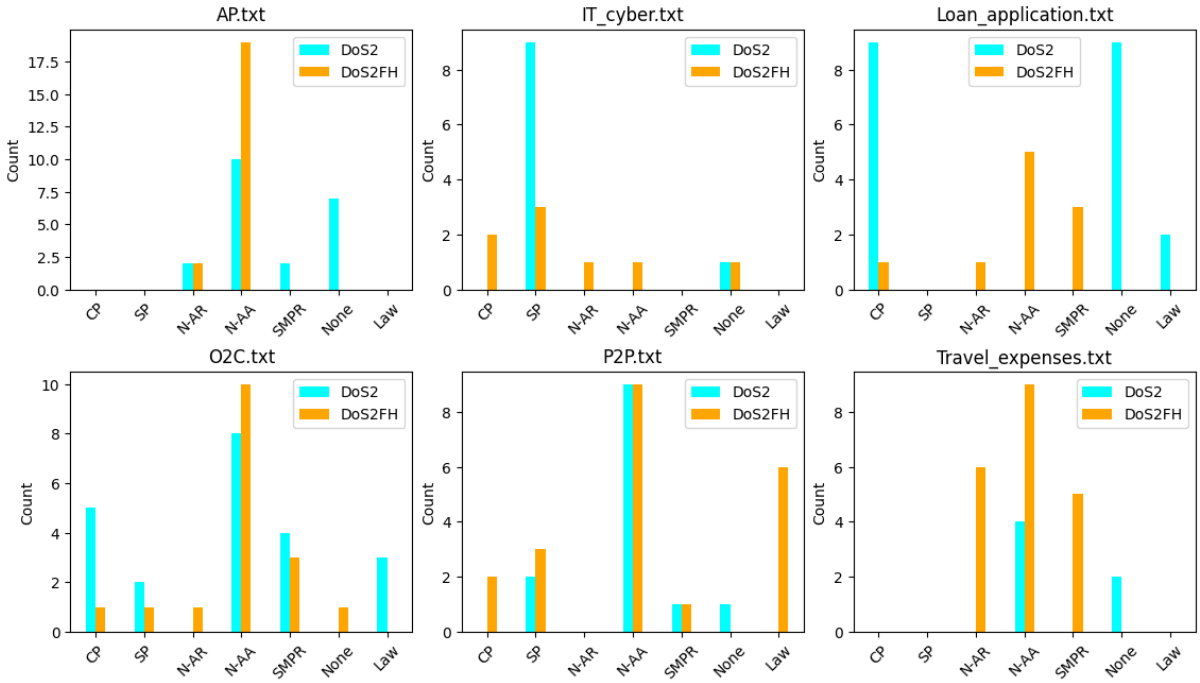

Figure 9: Results of references analysis for DoS2 and DoS2FH setup

6.3 Summary of results

The results show that increasing the degree of specialization to two has a positive contribution to the ability to answer process risk and process inefficiency questions. Increasing the degree of specialization to five did not have a statistically relevant positive contribution. Increasing the specialization by implementing human feedback was not useful in terms of output improvement, but there are qualitative advantages of user friendliness and transparency.

7 Discussion

The main question is:*to what extent do interactive and specialized DKGAs improve responses to process risk and inefficiency questions compared to a single general agent?* To answer the main question, the results of this research are analysed, followed by an overview of known limitations and future research perspectives.

7.1 Analysis of results

There are three aspects of the results that will be analysed, they are: the shortness of the textual output, the decrease in contribution percentage of DoS4 compared to DoS2, and the None count in the references. This is followed by a summary of the results.

In the main experiment, the DoS2 setup produces concise reports with few references. One could argue that, for the sake of the workflow output, a more concise report without repetition of arguments and with a strict reference selection is qualitatively better.

The DoS4 setup does not show a statistically relevant improvement in contribution percentage compared to DoS0. This may be because the writing agent is confronted with a large amount of information that needs to be integrated into the result, making it easier to lose sight of the main question. An example of this mentioned in the qualitative results, are two outputs where the company agent information were left out. This resulted in much lower contribution percentages in the DoS4 setup with General Electric and Google as the company variables.

An important discussion point in the references analysis is about the None count in the DoS2 setup. The Tavily search tool returned a series of links to non-existent webpages, especially in the processes of loan application and IT incident handling. On the contrary, the DoS2FH setup, which uses the GPT_researcher search tool, is much better at gathering existing references. GPT_researcher is a more effective search tool because it is an agentic setup that checks the references before they are passed onto the DKGA.

7.2 Known limitations

There are six important limitations in this research. They are: the variation in setup of the Flow, the sequential order of agents, the prompting knowledge necessity, the rigid role-specialization to answer the questions, the simplicity of the setups and the reliance on OpenAI models.

The variation in setup for the Flow setup may have been a contributing factor to the lower contribution percentage. This is an unwanted outcome because only the human feedback implementation method should have had an effect on this.

Having a sequential order of agents in all the setups meant that there was less agentic "coordination" than expected from a multi-agent system. Theoretically speaking, the execution order of the DKGAs should not affect the output too much because they have their own goals. However, it cannot be automatically assumed that the same behaviours are exhibited in a non-sequential multi-agent system.

The qualitative analysis in the human-in-the-loop experiment explains that prompting knowledge is needed to ensure correct feedback implementation. Conceptually, the program should be accessible for people without domain knowledge at hand, so this contradicts with the computer science background needed to provide feedback. Of course, this issue may be worked around by having a guide for the user feedback, but it does not remove the issue.

The rigid role-specialization of DKGAs makes the experimental setups limited to only the six questions made for this research. To be able to expand upon this research with different questions, it is necessary to change the role prompting of the DKGAs, and perhaps also the number of DKGAs.

Because DoS0 is a simplified setup of a previous research by Vogt et al. (2024), there is a possibility that results found in the two experiments conducted are not reproducible on the previous setup.

Only OpenAI models are used for the setups. Although in this research, this was the default for many of the used tools and frameworks, the writing agent's functionality depends heavily on the choice of model. Using a different model might have yielded better results.

7.3 Future work

Based on the three limitations - variation in the Flow setup, the sequential order of agents, and the rigid role-specialization to answer the questions, three promising aspects of future work can be

identified. They are: evaluating the difference between search engine tools, testing more permutation orders of DKGAs, and automatically generating the desired amount of role-prompted DKGAs.

The different Crew and Flow reference gathering methods greatly affected the quality of the references and, therefore, the output. Future research could experiment with utilizing different reference-gathering tools and measuring the effect of the workflow output, perhaps even creating a benchmark for certain tasks and certain desired reference gathering tools. Token limits and ease of use can be variables in this future research.

All the agents in the five setups of this research are run sequentially in one particular order. A more thorough experiment could test all possible permutation orders of DGKAs. For example, DoS2 has three DKGAs: a process agent (P), an agent with a focus on a particular risk or inefficiency (F), and a company agent (C), therefore, there are six possible permutations of the DKGAs $\{P, M, C\}$. Rerunning the setups with different permutations of agents can help to prove whether the setup would work for non-sequential multi-agent systems. This does not take into account that in a multi-agent system, agents can be run multiple times. However, testing all permutations when the agents can rerun is not viable.

The most promising future research could look at automatically generating the desired amount of role-prompted DKGAs in a setup when provided with a certain (process mining) question. The setups were tested on six questions, which were all derived from the same question template. This research used three variables in the question template: focus, process, and company, and the optimal degree of specialization was two. To be able to achieve this, it is important to do a preliminary test to see whether increasing the amount of variables in the question also increases the optimal degree of specialization. Perhaps, even a function could be created to mathematically represent an estimation of the optimal amount of DKGAs per presented question.

8 Conclusion

Increasing the degree of specialization of agents in sequential multi-agent systems proves effective for answering process risk and process inefficiency questions. The implementation of interaction through user feedback adds transparency and user friendliness to the system, but does not improve the question answering.

The main experiment focused on increasing the amount of Domain Knowledge Gathering Agents (DKGAs) and assigning specializations with role-based prompting. Increasing from a setup with one DKGA to three DKGAs (DoS2) had an average contribution percentage increase of 17.26%. This was a statistically significant increase according to the Wilcoxon signed-rank test; thus, the null hypothesis - there is no difference in performance between DoS0 and DoS2 - was rejected. However, when running a setup with five DKGAs (DoS4), the writing agent struggled to combine outputs from all DKGAs, and the average contribution percentage increased by 7.20% compared to one DKGA. This was not a statistically significant increase according to the Wilcoxon test.

A human-in-the-loop experiment provided the DKGAs with the human trait of interaction by allowing user feedback. The user friendliness and system transparency increased; however, the average contribution percentage of the textual outputs did not increase. User-friendliness was demonstrated through system interactivity, while the transparency was shown in the user's ability to view all intermediate outputs from the DKGAs. This allowed for catching errors before the outputs were combined into a final report. Of the two human-in-the-loop setups, the within-agent setup

(DoS2CH) had only a 4.08% increase in average contribution percentage, and the between-agent setup (DoS2FH) had a decrease in average contribution percentage. The decrease was possibly due to the different architecture, reference gathering method, and framework used. The reference gathering method of the between-agent setup was more effective in gathering existing references that linked to webpages.

Six limitations of the setups were discussed, they are: the variation in reference gathering tools between the Crew and Flow setups, the sequential order of agents, the prompting knowledge necessity, the rigid role-specialization, the simplicity of the setups, and the reliance on OpenAI models.

Three suggestions for future work were given. Firstly, rerunning the experiment on different sequential orders of agents can see whether the setup would work for non-sequential multi-agent systems. Secondly, because the difference in reference-gathering tools had a large effect on the selection of references, it is useful to compare reference-gathering tools to measure the difference in reference quality. The most promising future research would be to build upon the findings of this work to create an automatically generated number of role-specialized DKGAs, depending on the complexity of the process mining questions. To achieve this, the effect of different amounts of question variables in relation to the optimum DoS number should be researched.

References

Acharya, D., Kuppan, K., & Divya, B. (2025). Agentic AI: Autonomous intelligence for complex goals—A comprehensive survey. *IEEE Access*, *13*, 18912–18936. https://doi.org/10.1109/ACCESS.2025.3532853

Amirkhani, A., & Barshooi, A. H. (2022). Consensus in multi-agent systems: A review. *Artificial Intelligence Review*, *55*. https://doi.org/10.1007/s10462-021-10097-x

Barbarroxa, R., Gomes, L., & Vale, Z. (2025). Benchmarking large language models for multi-agent systems: A comparative analysis of AutoGen, CrewAI, and TaskWeaver. In P. Mathieu & F. De la Prieta (Eds.), *Advances in practical applications of agents, multi-agent systems, and digital twins: The paams collection* (pp. 39–48). Springer Nature Switzerland.

Berti, A., Kourani, H., & van der Aalst, W. M. P. (2025). Pm-llm-benchmark: Evaluating large language models on process mining tasks. In A. Delgado & T. Slaats (Eds.), *Process mining workshops* (pp. 610–623). Springer Nature Switzerland.

Berti, A., Maatallah, M., Jessen, U., Sroka, M., & Ghannouchi, S. (2024). Re-thinking process mining in the AI-based agents era. https://arxiv.org/abs/2408.07720

Berti, A., van Zelst, S., & Schuster, D. (2023). Pm4py: A process mining library for python. *Software Impacts*, *17*, 100556. https://doi.org/10.1016/j.simpa.2023.100556

Bertrand, Y., Weerdt, J. D., & Serral, E. (2024, February). An expert-validated bridging model for iot process mining - business & information systems engineering. https://link.springer.com/article/10.1007/s12599-023-00849-0#citeas

Bhaskar, H. L., & Osama, M. (2025). Maximizing business process efficiency in industry 4.0: A techno-functional exploration of process mining tools. *Operations Research Forum*. https://doi.org/10.1007/s43069-025-00428-x

Buchanan, J., Hill, S., & Shapoval, O. (2024). ChatGPT hallucinates non-existent citations: Evidence from economics. *The American Economist*, *69*(1), 80–87. https://doi.org/10.1177/05694345231218454

Chelli, M., Descamps, J., Lavoué, V., Trojani, C., Azar, M., Deckert, M., Raynier, J.-L., Clowez, G., Boileau, P., & Ruetsch-Chelli, C. (2024). Hallucination rates and reference accuracy of ChatGPT and Bard for systematic reviews: Comparative analysis. *J Med Internet Res*, *26*, e53164. https://doi.org/10.2196/53164

Chen, P., Zhang, S., & Han, B. (2024, June). CoMM: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Findings of the association for computational linguistics: Naacl 2024* (pp. 1720–1738). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-naacl.112

Conforti, R., Dumas, M., La Rosa, M., Maaradji, A., Nguyen, H., Ostovar, A., & Raboczi, S. (2015). Analysis of business process variants in apromore. *CEUR Workshop Proceedings*, *1418*, 16–20.

Dhurandhar, A., Nair, R., Singh, M., Daly, E., & Natesan Ramamurthy, K. (2024, August). Ranking large language models without ground truth. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Findings of the association for computational linguistics: Acl 2024* (pp. 2431–2452). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.143

Fontenla-Seco, Y., Winkler, S., Gianola, A., Montali, M., Lama, M., & Bugarín-Diz, A. (2023). The droid you're looking for: C-4PM, a conversational agent for declarative process mining. *Proceedings of the 21st International Conference on Business Process Management (BPM 2023), Demos & Resources Forum*, 112–116. https://ceur-ws.org/Vol-3469/paper-20.pdf

Getu, T. M., Kaddoum, G., & Bennis, M. (2024). A survey on goal-oriented semantic communication: Techniques, challenges, and future directions. *IEEE Access*, *12*, 51223–51274. https://doi.org/10.1109/ACCESS.2024.3381967

Gomes, A., Wanzeller, C., & Fialho, J. (2022). Comparative analysis of process mining tools. *CAPSI 2021 Proceedings*. https://aisel.aisnet.org/capsi2021/4

Gong, X., Liu, M., & Chen, X. (2024). Large language models with knowledge domain partitioning for specialized domain knowledge concentration.

Hutsebaut-Buysse, M., Mets, K., & Latré, S. (2022). Hierarchical reinforcement learning: A survey and open research challenges. *Machine Learning and Knowledge Extraction*, *4*(1), 172–221. https://doi.org/10.3390/make4010009

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, *55*(12), 1–38. https://doi.org/10.1145/3571730

Joshi, S. (2025). Advancing innovation in financial stability: A comprehensive review of ai agent frameworks, challenges and applications. *World Journal of Advanced Engineering Technology and Sciences*, *14*(2), 117–126.

Kong, A., Zhao, S., Chen, H., Li, Q., Qin, Y., Sun, R., Zhou, X., Zhou, J., & Sun, H. (2024). Self-prompt tuning: Enable autonomous role-playing in LLMs. https://arxiv.org/abs/2407.08995

Kourani, H., Berti, A., Schuster, D., & van der Aalst, W. M. P. (2024). Process modeling with large language models. In *Enterprise, business-process and information systems modeling* (pp. 229–244). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-61007-3_18

Krüger, A. K., & Petersohn, S. (2022). 'i want to be able to do what i know the tools will allow us to do': Practicing evaluative bibliometrics through digital infrastructure. *Research Evaluation*, *31*(4), 475–485. https://doi.org/10.1093/reseval/rvac009

Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., Wang, J., Chowdhury, T., Li, Y., Cui, H., Zhang, X., Zhao, T., Panalkar, A., Mehta, D., Pasquali, S., Cheng, W., Wang, H., Liu, Y., Chen, Z., Chen, H., ... Zhao, L. (2024). Domain specialization as the key to make large language models disruptive: A comprehensive survey. https://arxiv.org/abs/2305.18703

Malfa, E. L., Malfa, G. L., Marro, S., Zhang, J. M., Black, E., Luck, M., Torr, P., & Wooldridge, M. (2025). Large language models miss the multi-agent mark. https://arxiv.org/abs/2505.21298

Matthew Lam, L. H., Thatikonda, R. K., & Shareghi, E. (2024, December). A closer look at tool-based logical reasoning with LLMs: The choice of tool matters. In T. Baldwin, S. J. Rodríguez Méndez, & N. Kuo (Eds.), *Proceedings of the 22nd annual workshop of the australasian language technology association* (pp. 41–63). Association for Computational Linguistics. https://aclanthology.org/2024.alta-1.4/

Moed, H. F. (2005). *Citation analysis in research evaluation*. Springer.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., & Schulman, J. (2022). WebGPT: Browser-assisted question-answering with human feedback. https://arxiv.org/abs/2112.09332

Nguyen, H., & Litman, D. (2018). Argument mining for improving the automated scoring of persuasive essays. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). https://doi.org/10.1609/aaai.v32i1.12046

Plaat, A., van Duijn, M., van Stein, N., Preuss, M., van der Putten, P., & Batenburg, K. J. (2025). Agentic large language models, a survey. https://arxiv.org/abs/2503.23037

Rahman, A., Cvetkovic, V., Reece, K., Walters, A., Hassan, Y., Tummeti, A., Torres, B., Cooney, D., Ellis, M., & Nikolopoulos, D. S. (2025). MARCO: A multi-agent system for optimizing HPC code generation using large language models. https://arxiv.org/abs/2505.03906

Raina, V., Liusie, A., & Gales, M. (2024). Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2024.emnlp-main.427

Saravanos, A., Zervoudakis, S., Zheng, D., Nanda, A., Shaheen, G., Hornat, C., Konde Chaettle, J., Yoda, A., Park, H., & Ang, W. (2022). Reputation, risk, and trust on user adoption of internet search engines: The case of DuckDuckGo. In C. Stephanidis, M. Antona, S. Ntoa, & G. Salvendy (Eds.), *Hci international 2022 – late breaking posters* (pp. 683–691). Springer Nature Switzerland.

Satyadhar, J. (2025). Review of autonomous systems and collaborative AI agent frameworks. *International Journal of Science and Research Archive*, *14*(2), 961–972.

Schmidgall, S., Su, Y., Wang, Z., Sun, X., Wu, J., Yu, X., Liu, J., Liu, Z., & Barsoum, E. (2025). Agent laboratory: Using LLM agents as research assistants. https://arxiv.org/abs/2501.04227

Singh, A., Ehtesham, A., Kumar, S., & Khoei, T. T. (2024). Enhancing ai systems with agentic workflows patterns in large language model. *2024 IEEE World AI IoT Congress (AIIoT)*, 527–532. https://doi.org/10.1109/AIIoT61789.2024.10578990

Soni, A. B., Li, B., Wang, X., Chen, V., & Neubig, G. (2025). Coding agents with multimodal browsing are generalist problem solvers. https://arxiv.org/abs/2506.03011

Tripathi, A., Rai, A., Singh, U., Vyas, R., & Vyas, O. P. (2024). Unveiling ai efficiency: Loan application process optimization using pm4py tool. In D. Garg, J. J. P. C. Rodrigues, S. K. Gupta, X. Cheng, P. Sarao, & G. S. Patel (Eds.), *Advanced computing* (pp. 490–499). Springer Nature Switzerland.

Ugarte, M., Valle, P., Parejo, J. A., Segura, S., & Arrieta, A. (2025). ASTRAL: Automated safety testing of large language models. *Proceedings of the 2025 International Symposium on Software Testing and Analysis (ISSTA) – AST Companion Track*, 31–35. https://doi.org/10.1145/3713081.3731733

van Dongen, B. F., de Medeiros, A. K. A., Verbeek, H. M. W., Weijters, A. J. M. M., & van der Aalst, W. M. P. (2005). The ProM framework: A new era in process mining tool support. In G. Ciardo & P. Darondeau (Eds.), *Applications and theory of petri nets 2005* (pp. 444–454). Springer Berlin Heidelberg.

Vogt, M., van der Putten, P., & Reijers, H. A. (2024). Providing domain knowledge for process mining with ReWOO-based agents [Held at ICPM 2024]. *Proceedings of the First International Workshop on Generative AI for Process Mining (GENAI4PM 2024)*.

Vomberg, A., de Haan, E., Nicolai, E. F., & Broekhuizen, T. (2024). Digital knowledge engineering for strategy development. *Journal of Business Research*, *177*, 114632. https://doi.org/10.1016/j.jbusres.2024.114632

Wu, J., Zhu, J., & Liu, Y. (2025). Agentic reasoning: Reasoning LLMs with tools for the deep research. https://arxiv.org/abs/2502.04644

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., & Wang, C. (2023). AutoGen: Enabling Next-Gen LLM applications via Multi-Agent conversation. https://arxiv.org/abs/2308.08155

Xu, B., Peng, Z., Lei, B., Mukherjee, S., Liu, Y., & Xu, D. (2023). ReWOO: Decoupling reasoning from observations for efficient augmented language models. https://arxiv.org/abs/2305.18323

Zerbato, F., Soffer, P., & Weber, B. (2022). Process mining practices: Evidence from interviews. In P. Delias, D. Fahland, & D. Fahland (Eds.), *Business process management: 20th international conference, bpm 2022, münster, germany, september 11–16, 2022, proceedings* (pp. 268–285). Springer. https://doi.org/10.1007/978-3-031-16103-2_19

Zhang, K., Zhang, H., Li, G., Li, J., Li, Z., & Jin, Z. (2023). ToolCoder: Teach code generation models to use API search tools. https://arxiv.org/abs/2305.04032

Zheng, M., Pei, J., & Jurgens, D. (2023). Is "a helpful assistant" the best role for large language models? a systematic evaluation of social roles in system prompts. *CoRR*, *abs/2311.10054*. https://doi.org/10.48550/arXiv.2311.10054

Zimmermann, L., Zerbato, F., & Weber, B. (2024). What makes life for process mining analysts difficult? a reflection of challenges. *Software and Systems Modeling*, *23*, 1345–1373. https://doi.org/10.1007/s10270-023-01134-0

9 Appendix

9.1 Side study: A critical evaluation LLM workflow outputs

9.1.1 Abstract

This side study explores the usefulness of different evaluation methods for LLM workflow outputs. Often, evaluations performed on textual workflow outputs are brief, or the definition of the selection criteria is not well-defined. The main research (about digitizing human expertise in process mining) created setups inspired by the previous work of Vogt et al. (2024). In this previous work, it was stated that more evaluation was needed besides the ablation test and the percentage of integrated domain knowledge (%IDK) measured by one person.

Three evaluation methods are thoroughly tested on the agentic research by Vogt et al. (2024) to test the effectiveness. The three tested evaluation methods are calculating the %IDK with multiple graders, a reference analysis, and a Large Language Model-as-a-Judge (LLM-as-a-judge). The LLM-as-a-Judge method was used to replace the manual calculation of the percentage of integrated domain knowledge and the manual reference analysis.

The results of the %IDK method with human graders shows that the human graders had various interpretations of the definition used for the %IDK which led to a large standard deviation of answers. However, the method was reliable in terms of explainability.

The results of the references analysis method are a classification for reference gathering and a qualitative analysis. The classification is a table with seven categories for references, these can indicate the quality of the references.

The Large Language Model-as-a-judge method for replacing the %IDK resulted in a worse classification compared to the human %IDK grading method and the human classigication method.

Future research would benefit from creating a new quantitative metric to judge agentic outputs that has much stricter guidelines than the %IDK defintion.

9.1.2 Introduction

Commonly found limitations in the field of evaluating LLM outputs are that it requires more evaluation (Fontenla-Seco et al., 2023; Kourani et al., 2024; Vogt et al., 2024). This research will test some promising evaluation metrics for textual LLM outputs and take successful metrics to the main study, as well as tips for improvements of the evaluation methods.

The background research provides an oversight of commonly used evaluation metrics. In the methods section, the percentage of integrated domain knowledge (%IDK) and a references analysis are chosen due to their quantitative nature. The LLM-as-a-judge method was chosen because of its potentially promising nature to save time. The main question of this research is "what are the evaluation methods for textual LLM outputs that should be used for the main research?".

The coding for this side study can be found on GitHub

9.1.3 Background and related work

The grading of textual reports has elements of subjectivity, which can be eliminated once there is a ground truth. This background research will look at one evaluation method, a references classification, which is able to make part of the textual output have a ground truth. Four other

evaluation methods will be introduced which do not introduce a ground truth but they are useful for a qualitative and/or quantitative analysis. These methods are: user validations with experts, ablation testing, the %IDK metric and the LLM-as-a-judge.

A reference classification is a methodology of the "evaluative bibliometrics" field. This field is "aimed to construct indicators of research performance from a quantitative analysis of scholarly documents"(Moed, 2005). There is much discussion that surrounds the reference classification of scientific papers (Krüger & Petersohn, 2022), but not much on a reference analysis of a paper with different origins, such as blogs from Google, social media posts, and company research papers. In a research done by Buchanan et al. (2024) on hallucinations of ChatGPT in an Economics context, 30% or more of the citations were non-existent. This is what a reference analysis can bring to the table.

Expertise evaluations are often performed to check whether a model or a setup creates an output that adheres to the standards of domain experts. An example work which created a "bridging model for IoT process mining" (Bertrand et al., 2024) and used an expert panel to quantitatively grade the quality of the output. They had 12 experts who performed the analysis. The method of evaluation proved useful to prove the effectiveness of their created model.

Ablation testing is a system decomposition. Within the area of computer science, it is used to test the usefulness of program components. An example in process mining research is from Vogt et al. (2024), which used ablation testing on his created Proof of Concept (PoC). The ablation testing took the process mining component, the domain knowledge gathering component and the whole PoC. This method was useful for showing the added value of the PoC. A research by Nguyen and Litman (2018) used ablation testing on features to test their model for argument mining in automated essay scoring, which was successful for showing the contribution of the features.

The percentage of integrated domain knowledge measures the presence of company-specific and process-specific knowledge with backed-up references. This metric is taken from a process mining research (Vogt et al., 2024) of which the output is similar to that of the main research of this paper. The definition gathered from the research is: "the content represented [...] external ecosystem domain knowledge specific to the process component(s)". Because it is a quantitative metric, it is useful for paper comparison.

An interesting evaluation method is the LLM-as-a-judge (Berti et al., 2025; Dhurandhar et al., 2024) or "automated reviewer" as other research words it (Schmidgall et al., 2025). Previously, this method yielded positive results in a research by Berti et al. (2025) which "justified" the used LLM-as-Judge method in their results. The research does emphasize that it is important to select "advanced LLMs for the task". Additionally, in the research by Schmidgall et al. (2025), "the automated reviewer achieved human-level accuracy ... and surpassed human performance in F1 score". A paper that analysed Zero-Shot LLM Assessment stated that the method had many advantages, such as being very similar to human judgement, the ability to evaluate texts without "domain-specific training" and the straightforwardness of the evaluation. However, it also mentions that different forms of bias are a large risk; this is something to be aware of during these experiments (Raina et al., 2024).

LLM-as-a-judge can also be used as an evaluation method to cite references. Despite the same advantages of grading textual outputs, a hallucination factor has to be taken into account when letting the LLM grade references. A recent paper by (Chelli et al., 2024) resulted in a GPT-4 accuracy score of just 13.4% when gathering references. The advice that followed was that references should always be checked by a human. Not only the citations, but also the contents of the references

can be hallucinated, especially when using "lower performing models, such as GPT-4o" (Schmidgall et al., 2025).

9.1.4 Method

This section first selects the chosen evaluation methods from the background research, and then it prepares the textual outputs for evaluation. Followed by a general evaluation setup and an in-depth explanation of the evaluation method.

Selecting the chosen methods from the background research

Although a reference classification is not often used in process mining, it will be used for the evaluation because it is useful to determine whether references are hallucinated and what the origins of the references are.

An expertise evaluation will not be conducted because to evaluate the papers from the model, at least twelve experts would be needed from different fields. This is because of the six questions answered, there are thirteen different expert fields present. For example, the question "Can you find the **environmental risks** in the **purchase to pay (P2P) process** at **Walmart** and what are the potential causes for the environmental risks" has at least four different experts that can give their opinion on the output. The fourth expert is a process mining expert who reads whether the abstraction of the event log that is analysed is correctly integrated into the report to answer the question.

An ablation test was already performed on the paper in question. Within the main experiment of this research, there is a choice to implement this method on the main experiment of desired.

Since the setups of the main research in this paper are based on the setup from this paper, the %IDK is a useful metric to analyse. In the background section, it was mentioned that the performed evaluation was done by only one human grader. In this evaluation, four human graders will evaluate the outputs.

Due to the promising nature of the LLM-as-a-judge method, this side study will test two different implementations of the LLM-as-a-judge method. One is an automatic %IDK estimator, and the other is an automatic references categorization.

Preparing textual outputs for evaluation

The table of contents was removed from all output papers because only qualitative information from the report should be taken into account, whereas the table is a repetition of the titles of chapters. In the definition of %IDK it is undefined when the chapter title is highlighted in the report, whether this should also be the case for the title chapter in the table of contents.

General evaluation setup

This side study is conducted in four small experiments. This will use the %IDK evaluation method with multiple human graders, the %IDK evaluation method with LLM-as-a-judge, a human references analysis, and an LLM-as-a-judge references analysis:

1. Calculating the %IDK implementation with human graders
2. Calculating the %IDK implementation with LLM-as-a-judge
3. Human references analysis

Problem identification	Process	Company	Run	Filename
Environmental risk	Purchase-to-Pay	Walmart	1	P2P_1
“ ”	“ ”	“ ”	2	P2P_2
Audit risk	Order-to-Cash (O2C)	Procter and Gamble	1	O2C_1
“ ”	“ ”	“ ”	2	O2C_2
IT & cyber risk	IT incident handling (INC)	Volvo	1	INC_1
“ ”	“ ”	“ ”	2	INC_2
Process inefficiencies	Accounts Payable (AP)	General Electric (GE)	1	AP_1
“ ”	“ ”	“ ”	2	AP_2
Operational risk	Travel expenses	Google	1	EXP_1 or TE_1
“ ”	“ ”	“ ”	2	EXP_2 or TE_2
Regulatory risk	Loan application	Wells Fargo Bank	1	LA_1
“ ”	“ ”	“ ”	2	LA_2

Table 9: Textual outputs explanation

4. LLM-as-a-judge references analysis

Within these experiments, twelve textual outputs are graded[33]. These outputs are the outputs of six experiments, run twice. An explanation of the topics of the six experiments and their abbreviations which will be used in the results, can be shown in Table 9.

In-depth explanation per evaluation method

The four in-depth explanations for the %IDK, the %IDK with LLM-as-a-judge, the reference analysis and the reference analysis with LLM-as-a-judge are given below.

Calculating the %IDK implementation with human graders. Four human evaluators were selected to highlight the %IDK of the textual outputs. All human evaluators had different scientific backgrounds in Biology, Mathematics and Business. All four raters were shown the same introductory presentation to explain the definition in detail and they also saw some examples separately. The %IDK is automatically calculated with a simple Python program that was written especially for this[34].

This experiment will be followed by the %IDK analysis utilizing LLM-as-a-judge while making use of zero-shot prompting. The prompt of the LLM is:

```
f''Given the following question: {specific_file_question}
And the answer to the question:
{answer_text}
Please evaluate the above answer on the amount of
specific external ecosystem domain knowledge.
It is specific external ecosystem domain knowledge
when it is linked to the company or the step being
analyzed within the company and process mining
domain knowledge.  Return a percentage of the text
```

[33]https://drive.google.com/drive/folders/1jm7CRsY21WP1OQ_64dxk_4gik-ZHPE47

[34]https://github.com/CodingBuddy25/Evaluation_LLM/blob/main/main.py

```
that has specific external ecosystem
domain knowledge."
```

In this context, {specific_file_question} is a Python dictionary that elaborates per file what the specific question was. For example, for the Loan_application files it was:"Can you find the regulatory risks in the loan application process at Wells Fargo bank. What are the regulatory risks for specific steps that you identified?" {answer_text} is textual output without a table of contents and without the references, just like the human rater evaluation. Since an LLM may exhibit inconsistent results, the LLM-as-a-judge will be run ten times. The average and the standard deviations are taken from these ten measurements. The prompts and the code of which can be found on [35]

Following this, the references in the outputs will be evaluated as they are the foundation for the information in the paper. The results of both grading methods will be quantitative due to a categorization of the references.

Using the same categories as the human-as-a-judge evaluation, the references are classified using the LLM-as-a-judge. The partial prompt of the LLM is:

```
    ``Given the following references: {references}
    Please evaluate the references and put them into
    seven categories: [lists the 7 categories].
    The explanation per category is as follows:
    [explanation per category].

    TASK: Give the frequency, in a number per
    category. [task specifics] "
```

Whereby {references} is the file that is being analyzed. The full prompt[36] contains extra information and specifics. In practice, the references will be split up into batches of around 20 references per prompt to make sure the LLM-as-a-judge does not leave any references out.

9.1.5 Results

Quantitative evaluation of %IDK - human graders

Results from the experiment are shown in Table 10. Average scores of the P2P process (80.11% and 83.86%) are the highest in the table, and those for travel expenses (EXP) and loan application (LA) are the lowest on average. All averages are above 65% thus over 65% of the content represented domain knowledge specific to the process component(s).

Quantitative evaluation of integrated domain knowledge - LLM-as-a-judge

The results of this experiment are presented in the form of a box plot, see Figure 10.

As is clearly visualised in the box plots 10, the LLM is prone to having a couple of really low percentages, making some standard deviations really large. On the other hand, since the LLM-as-a-judge repeats the same experiment ten times, for many papers, the standard deviation is lower compared to the human raters. This is an advantage compared to the human rating method.

[35] https://github.com/CodingBuddy25/Evaluation_LLM/blob/main/LLM-as-a-judge.py
[36] https://github.com/CodingBuddy25/Evaluation_LLM/blob/main/LLM-as-a-judge.py

Table 10: %IDK by process and human rater

File	Evaluate_N	Evaluate_R	Evaluate_M1	Evaluate_M2	Average	Standard Deviation
AP_1	72.13	84.45	84.85	79.63	80.27	5.92
AP_2	62.57	73.08	63.33	80.06	69.76	8.37
INC_1	41.10	85.80	84.09	87.24	74.56	22.34
INC_2	57.86	93.39	82.06	89.24	80.64	15.89
LA_1	55.70	72.65	58.08	81.63	67.02	12.29
LA_2	61.65	65.93	78.76	73.83	70.04	7.70
O2C_1.	67.72	78.41	88.67	68.45	75.81	9.86
O2C_2	61.66	72.87	90.63	77.69	75.71	12.00
P2P_1	68.65	82.59	94.84	74.35	80.11	11.37
P2P_2	71.06	96.98	89.12	78.26	83.86	11.47
EXP_1	75.86	78.84	56.68	74.19	71.39	10.00
EXP_2	73.64	89.60	80.76	59.48	75.87	12.73

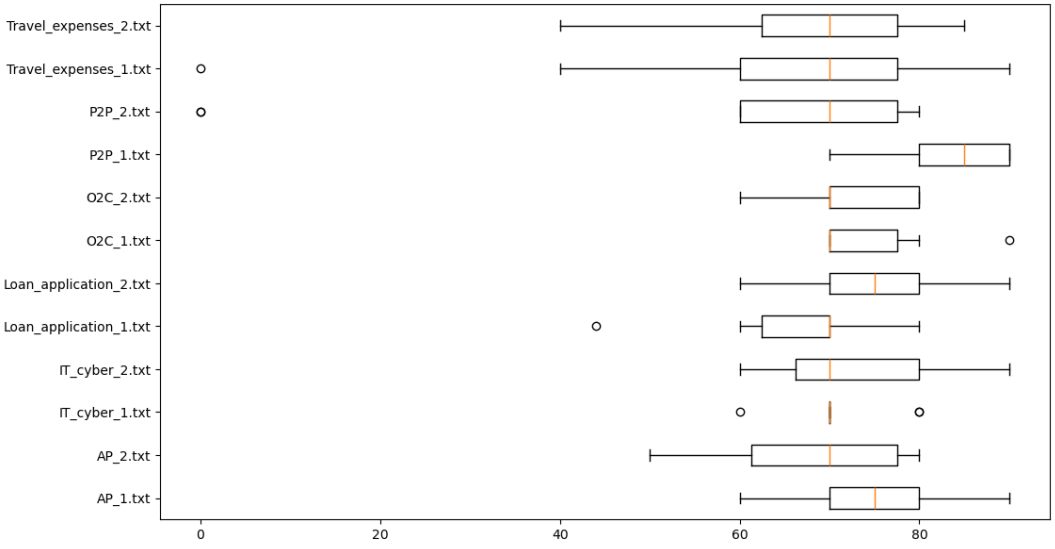

Figure 10: Average and standard deviation of %IDK by process (LLM-as-a-judge)

Different humans may interpret the relevant domain knowledge slightly differently, even though the same definition and explanation is given.

References analysis - human grader

In total, 766 references were classified by hand, resulting in a categorization of the references and a document of classified references. The categorization of the references can be found in Table 11. Effort was put into ensuring the categories were mutually exclusive and collectively exhaustive. The document of classified references can be found in the GitHub environment, the colour classification linked to the category can be found in the README.md file. A visual categorization of all the references into seven categories category in Figure 12 and per output paper in Figure 13. The visual categorization can be found in combination with the results from the LLM-as-a-judge method.

There are three quantitative results that are important to mention: the none count, the ratio of non-academic articles and the amount of domain-specific spikes. As can be seen in the Figure 12, the AP_2 output from the accounts payable analysis, the output has more than 20 hallucinated references. This is an anomaly compared to the other papers, which have between zero to five hallucinated references.

The ratio of non-academic articles is illustrated clearly in Figure 11 where most of the papers have significantly more references that are Non-AA than the other categories.

Finally, there are some domain-specific category spikes that can be found. Returning to Figure 12 shows that especially TE_1 and TE_2 contain many Law documents. IT_1, IT_2, P2P_1 and P2P_2 seem to use many scientific papers compared to the other outputs.

References analysis - LLM-as-a-judge

The human references analysis is considered to be the ground truth. With this, there are two interesting points of analysis: misclassification of references and amount of references classified.

According to the LLM-as-a-judge, law documents are present in the accounts payable and IT incident handling domains but this is not the case. LLM-as-a-judge often assigned None and social media references to different categories.

In terms of the amount of references classified, the LLM-as-a-judge did not always grade all the references, despite the fact that the references were split up into lists of 20, as was mentioned in the method. In the EXP_1 paper, the LLM-as-a-judge categorized more references than were given. This can be seen in the difference count of Figure 13.

9.1.6 Discussion

Analysis of %IDK as a human grading method

The human analysis was time consuming and as can be seen in the results, there is also a large variation between the graders, despite having the same definiton of %IDK.

This method was useful to analyse the outputs of large textual inputs quantitatively because scores between papers can be compared easily. For example, the fact that average scores for Audit risk and Environmental risk were higher than those for Process inefficiencies and Regulatory risk could be because is more domain knowledge in online sources available about the process in question.

Analysis of references human grading method

Categories	Description
Self-published company papers (CP)	The source is from a page of the company that is getting analyzed by the setup. For example, www.media.volvocars.com for the Volvo prompt (IT_cyber_1 and IT_cyber_2) and www.pgsupplier.com for the P&G prompt (O2C_1 and O2C_2).
Scientific publications (SP)	These references link to an acknowledged publications platform such as www.tandfonline.com, www.nature.com, and link.springer.com.
Non-academic research (Non-AR)	References that lead to research done by companies. Often these are large consulting companies that have done extensive market research. It cannot be, for example, a blog of one person who has done a small internet search. This definition should encapsulate research that has been done with significant effort, significant evidence or referencing, and/or a significant number of contributors.
Non-academic article (Non-AA)	Any source that is non-academic and that does not fall in the category above falls into this category. Often these are blogs of people that have done research into the definition or process in a particular domain.
Social media or press release (SM or PR)	This includes articles and news that is published on social media platforms or press platforms. References from social media platforms are often from LinkedIn, and press release platforms range from papers such as "The Guardian" to "Forbes."
Link does not exist (None)	This is a link that does not lead to an article. It may be that the link once existed and that the company has removed the article, or it may be that it has never existed at all.
Law documents	This is a small category made for law documents. These are official documents that factually state the laws. The domain of loan applications for the company Wells Fargo.

Table 11: Category explanations of references classification

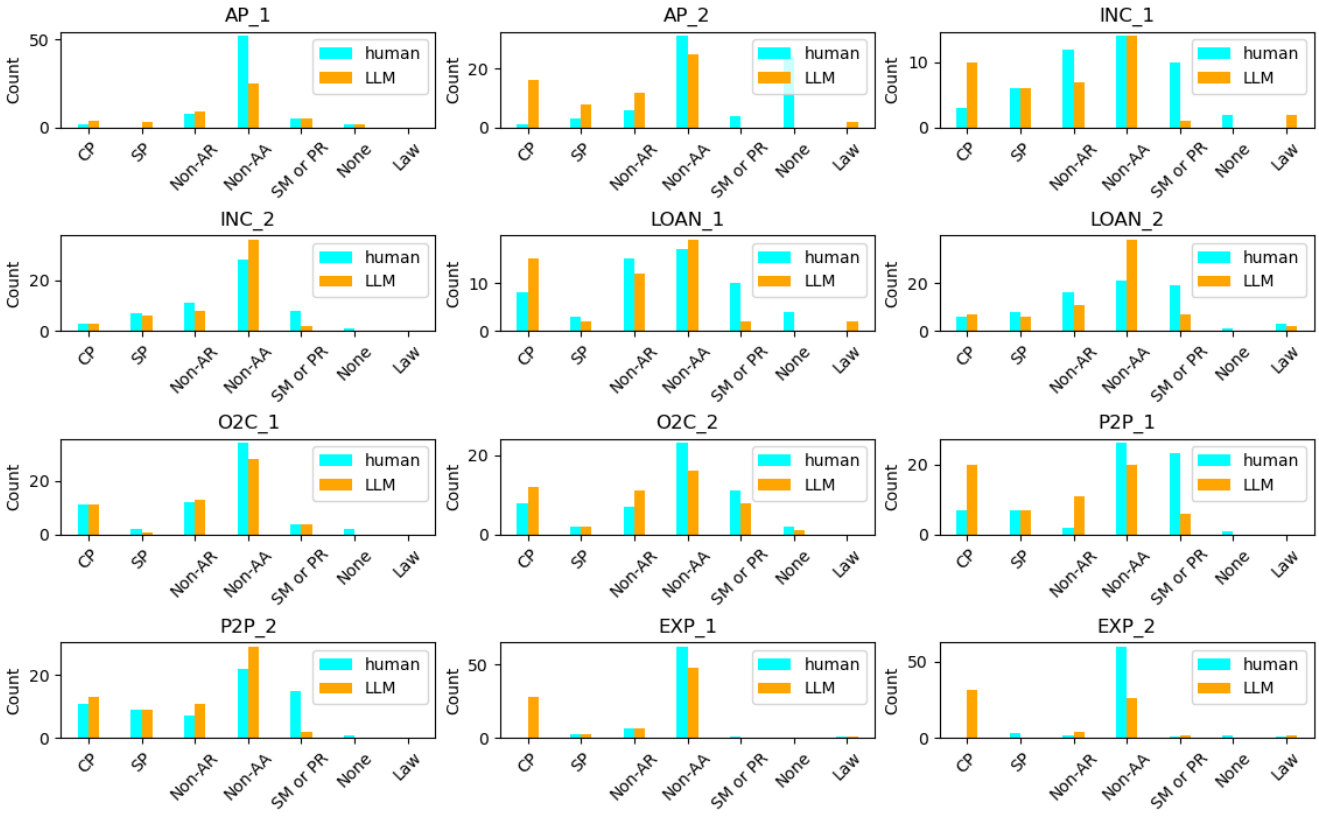

Figure 11: References classification in category count per paper

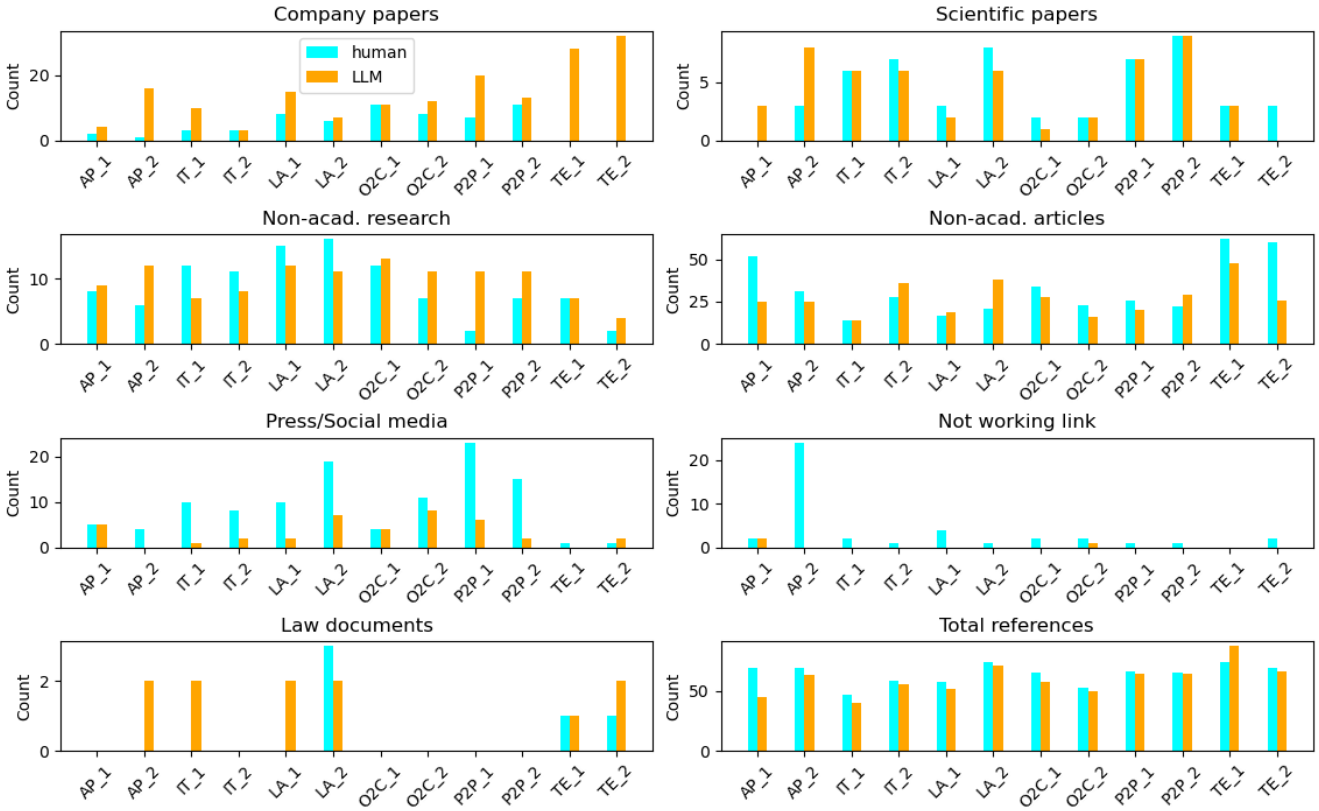

Figure 12: References classification in count per category

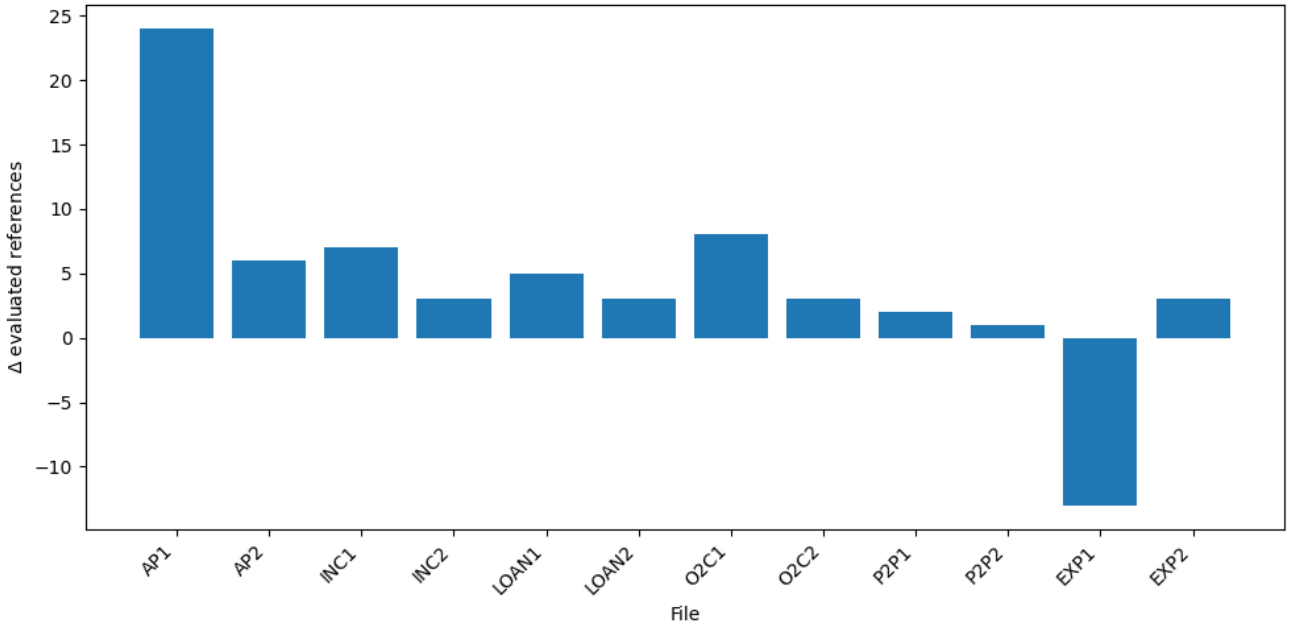

Figure 13: Difference in reference count per paper

Similarly to the human grading method, the manual reference grading process was time-consuming. However, one may argue that it was even more useful than the %IDK because the references are the foundations of the paper. For example, a paper with few working reference links also experienced lower IDK percentages (AP_2). Additionally, few references are scientific papers, it is possible that it is more opinionated rather than factually accurate. In the results, it is shown that the O2C papers do not perform as well, and they both happen to have very few scientific papers in the references.

Analysis of LLM-as-a-judge method

The LLM-as-a-judge method proved useful in the time-saving aspect but the results in terms of calculated percentage of IDK and references analysis are not fruitful. The largest issue of the LLM-as-a-judge is the large variation in percentages, as can be seen in the results for the %IDK calculation, it suffers from large outliers. In terms of the references classification, LLM-as-a-judge skipped and misclassified a substantial number of references.

Known limitations and future work

As mentioned in the method, the %IDK is not a defined metric in the paper that uses this. This method for quantifying textual outputs is useful, but it is important to create a clearer definition in future work.

The manual classification of references performed well. A limitation is that there is a fine line between non-academic research and non-academic articles, thus the decision is left to the human grader.

It is not recommended to use the LLM-as-a-judge method for classifying references due to the limitations in misclassifications and not grading all the references. However, the LLM-as-a-judge for the %IDK may have some future potential due to limitations of the zero-shot prompting used in this experiment. It might be beneficial to add a few-shot prompting method, possibly giving more positive results. Additionally, the LLM-as-a-judge suffered from large outliers. Future work could, instead of running the LLM-as-a-judge ten times, run it fifty times and remove the outliers, the results will be more promising. Especially when considering that the human grading also suffered from large standard deviations between the different graders, but rerunning those experiments many times is time-wise not achievable.

9.1.7 Conclusion

The three tested evaluation methods in this side study were: calculating the percentage of integrated domain knowledge (%IDK) with multiple graders, a reference analysis and a Large Language Model-as-a-Judge. The Large Language Model-as-a-Judge method was used to replace the calculation of the percentage of integrated domain knowledge and the reference analysis.

The human percentage of domain knowledge (%IDK) evaluation method was useful to quantify the output of the model. Different human graders led to different interpretations of the definition, which led to a large standard deviation. It is recommended to use one human evaluator which repeats the evaluation multiple times. This way, the outcome is more comparable and reliable.

The LLM-as-a-judge for identifying the %IDK was not useful because it led to even larger standard deviations than the human grading system. Also, the output was less explainable because there is no output of which presents the parts of the text which were counted towards the %IDK.

The human graded references analysis was useful as a metric, although it was time consuming.

The results of the human references analysis method are a classification for reference gathering and a qualitative analysis. The classification is a table with seven categories for references, these can indicate the quality of the references, they are: self-published company papers, scientific publications, non-academic research, non-academic articles, social media or press release, None, and Law documents. The LLM-as-a-judge for the references analysis is not recommended for use because references were skipped and often misclassified, despite the fact that prompts got broken up in sections of 20 references.

In future work, it is recommended to use a detailed and clear definition of such a metric quantitative metric as well as a manual references analysis.

9.2 Abstraction figures

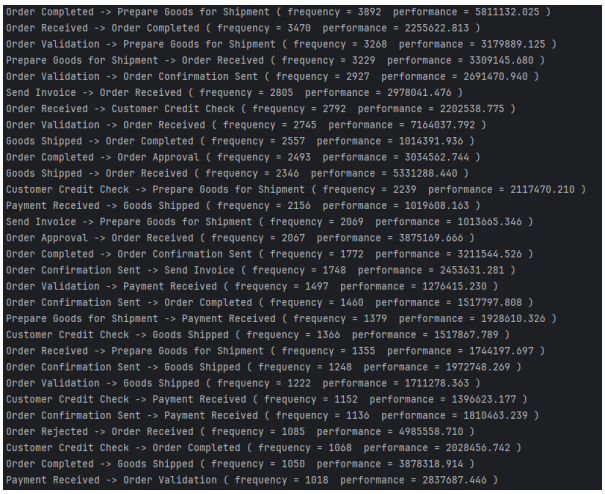

Figure 14: Example of DFG abstraction

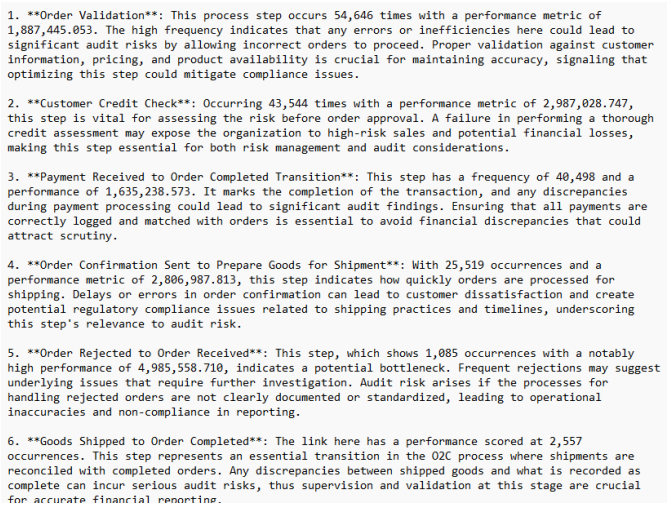

Figure 15: Example output process analysis

9.3 HITL Crew interface

Figure 16 is the human-in-the-loop interface for the Crew setup.

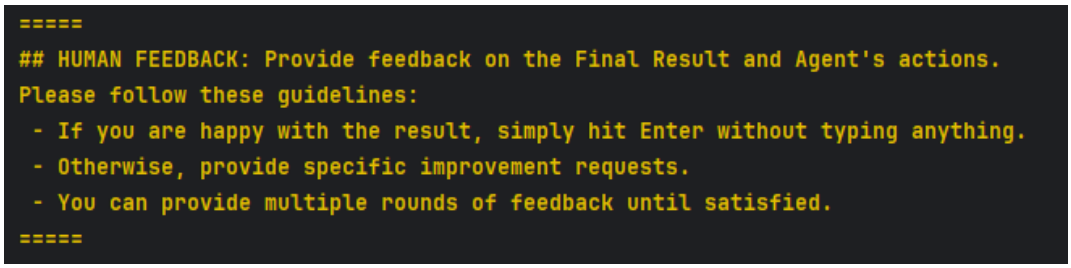

Figure 16: Human input interface

9.4 Tavily search result example

Every DKGA can gather external domain knowledge either through an official Tavily search or the LLM within the agent, which can access information.

As mentioned in the coding implementation in method Section 4.3, the Tavily search returns the source, title, and content. This summary is shown to the user when the program is running.

1. "Supply Chain Management Model of Procter and Gamble." UKEssays. [URL] (https://www.ukessays.com/essays/business/supply-chain-management-procter-gamble.php)

2. "Order to Cash (O2C) Process: A Complete Guide." HighRadius. [URL] (https://www.highradius.com/resources/Blog/order-to-cash-process-optimization/)

3. "Improvement of an order-to-cash business process by deploying..." Emerald Insight. [URL] (https://www.emerald.com/insight/content/doi/10.1108/ijppm-01-2022-0050/full/html)

4. "How a Collaborative O2C Process Eliminates Cash Flow Bottlenecks." Versapay. [URL] (https://www.versapay.com/resources/collaborative-order-to-cash-process-cash-flow-bottlenecks)

5. "P&G: End-to-end Supply Chain Model." Harvard Business Review. [URL] (https://d3.harvard.edu/platform-rctom/submission/pg-end-to-end-supply-chain-model/)

6. "What Is the Order-to-Cash Process and What Are its 8 Stages?" Conexiom. [URL] (https://conexiom.com/blog/what-is-the-order-to-cash-process-and-what-are-its-8-stages)

7. "Procter & Gamble's Operations Management: 10 Critical Decisions." Panmore Institute. [URL] (https://panmore.com/procter-gamble-operations-management-10-decisions-productivity)

8. "How the P&G Supply Chain Thrives on Supply Chain Complexity." Tools Group. [URL] (https://www.toolsgroup.com/blog/procter-gamble-supply-chain-complexity/)

9. "How P&G stays on top of its game." Supply Chain Management Review. [URL] (https://www.scmr.com/article/Insiders-view-of-Proctor-and-Gamble-supply-chain-success) 1

10. "How Complete and Timely Invoicing Can Reduce Delayed Payments." Payment Cycle. [URL] (https://www.paymentcycle.com/blog/invoicing-reduce-delayed-payments) 1

11. "The Importance of Accurate Data in the O2C Process." ZS Associates. [URL] (https://www.zs.com/insights/importance-of-accurate-data-in-order-to-cash) 1

12. "Analyzing O2C Process Performance Metrics." Finance Times. [URL] (https://www.financetimes.com/analyzing-o2c-process-performance/) 1

13. "Eliminating Manual Work in Order to Cash." Business Process Management Journal. [URL] (https://www.bpmjournal.com/eliminating-manual-work-in-order-to-cash) 1

14. "Streamlining Order to Cash with Automation." Automation World. [URL] (https://www.automationworld.com/business/article/21140700/streamlining-order-to-cash-with-automation)

15. "The Role of Customer Relationship Management in O2C." CRM Magazine. [URL] (https://www.destinationcrm.com/Articles/ReadArticle.aspx?ArticleID=74069)

16. "Bottlenecks in Order to Cash and How to Address Them." Supply Chain Quarterly. [URL] (https://www.supplychainquarterly.com/articles/54-bottlenecks-in-order-to-cash-and-how-to-address-them)

17. "Impact of Supply Chain Complexity on Cash Flow." Journal of Business Logistics. [URL] (https://www.journalofbusinesslogistics.com/)

18. "Reducing Errors in the Order to Cash Process." Business Process Review. [URL] (https://www.bprjournal.com/reducing-errors-in-order-to-cash-process)

19. "Enhancing Data Visibility in the O2C Process." Insights on Business. [URL] (https://www.insightsonbusiness.com/enhancing-data-visibility-in-o2c)

20. "Continuous Process Improvement in Order to Cash." Quality Management Journal. [URL] (https://www.qualitymanagementjournal.com/continuous-process-improvement-in-order-to-cash)

9.5 Results

9.5.1 Human Flow log file

Example event log that was made to be understandable for a human reader, it contains the agents that were called in the Flow and the exact timing that they were **finished**:

```
-Introduction agent: 2025-05-29 10:47:25.276898
-Process mining agent: 2025-05-29 10:48:10.851242
-Domain knowledge agent with company specialization: 2025-05-29 10:49:41
-Domain knowledge agent with process specialization: 2025-05-29 10:51:10
-Report writer agent: 2025-05-29 10:51:53.986709
-Report writer agent: 2025-05-29 10:54:45.681480
-Process mining agent: 2025-05-29 10:56:01.329373
-Report writer agent: 2025-05-29 10:56:53.473785"
```

9.5.2 HITL Flow diagram

This diagram is rendered when executing the code.

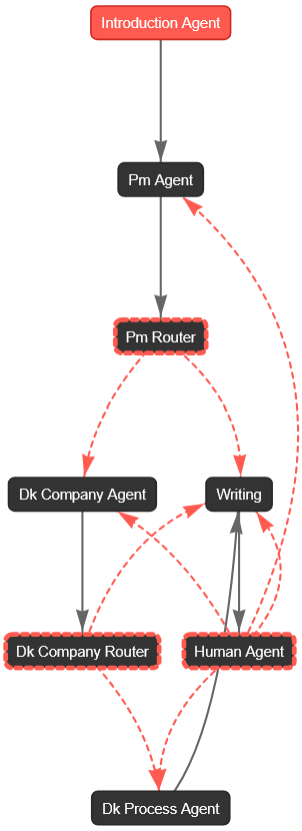

Figure 17: Rendered Flow diagram of setup