Research Project

German-Food Ontology and Q&A Knowledge Graph

Mateusz Frydryszak, B.Eng
Aqil

Faculty: Computer Science and Engineering
Esslingen University
Telkom University
August 2025 - September 2025

Period: 28.07.2025 - 07.09.2025
Reviewer: Dr. Kemas Rahmat Saleh Wiharja, M.Eng.
Second Reviewer: Aqil

Company:
Supervisor:

Acknowledgement

If you like, you can add a thanks to the profs, company and supervisor. Additionally, thanks to the family and friends :)

Abstract

As a first step towards your successful thesis, you need to evaluate a few things. First, is the topic of value for you and do you feel motivated to work on this project for either three or six months? This does not only mean that you work in a specific field for this period, but before you start, you already have a clear vision on how to start. Of course, things change over time, and it would be boring, if you would already know everything in advance. However, prepare yourself with research and discussions with your supervisor to be on top of the topic. Second, you need to document your thesis in a way that **people can follow your thoughts and understand decisions** you had to take on the way. This includes **clear writing** in a useful structure, where your supervisor will help you with. But it also includes the choice of language: English or German? (others are not accepted) Of course, English is preferred, as it fits best to the scientific environment where you will position yourself with this thesis. But you can still choose German to create a document without loosing time if your English is not fluent or too stable. In this document, you need to set the selected language at two different positions:

A) In the files *template/title.tex* and *template/metadata.tex*, you need to adjust the information on your thesis to generate a proper front page.

B) In *template/packages.tex* you need to set the correct option for the *babel* package by either using **\usepackage[english]{babel}** for english or **\usepackage[ngerman]{babel}** for german. If you want to use umlauts with e. g. packages like hyphenat you should use a font encoding with good support of accents. To do this, put **\usepackage[T1]{fontenc}** above the **\usepackage[ngerman]{babel}** command.

C) In *thesis.tex* you need to change the strings starting with "List of" to german and add **\renewcommand{\listalgorithmname}{Algorithmusverzeichnis}** above **\addcontentsline{toc}{chapter}{\listalgorithmname}**.

Contents

List of Figures

List of Tables

List of Listings

List of Algorithms

Todo list

1. Introduction

This LaTeXtemplate is just a guideline to get started with your thesis, it is not a one fits all solution. For further ideas and tips, please refer to additional literature, like "How to Write a Better Thesis" from Evans et al. [1].

1.1. Motivation

In most cases, your introduction will start with a motivation section. This section summarizes an existing problem and presents the environment for the solution. This environment also highlights the technical boundaries of your design and implementation.

1.2. Goals

After the motivation, you can shortly summarize the main goals of the thesis. These goals enable an evaluation that measures your implementation against these goals. For each goal, you can always highlight the evaluation technique to verify the achievement.

The following approach is fully optional and only recommended, if you have clear and distinct goals for your thesis. You can list the goals with an explicit label as follows.

Goal (G1) *This is an example goal to test the counter of the goal for correct references.*

Goal (G2) *Students should have a seamless start in writing the thesis, without unnecessary LaTeX troubles.*

You can reference the goals from anywhere in the document and always link decisions to original goals. For example, we write this document to fulfill Goal 2.

1.3. Structure

Finally, you should quickly highlight the structure of your thesis and the solution, but keep it short and leave out details. It is important that you already present a little insight into your structure, as the reader can follow easier through background and related work.

2. Background

The background section guides you through the use of the template and some features we included to make writing a little easier.

2.1. metadata.tex and title.tex

The file *template/metadata.tex* includes all information on you, the title of the thesis and supervisors. This information is used during the compilation of the template to fill out all the details. Especially in the file *thesis_template/title.tex*, you will see the use of this metadata. Please adjust this title page and the metadata to fit to your thesis topic, thesis category, company, and supervisors.

2.2. packages.tex

The file *template/packages.tex* includes all used packages. Please only add your packages here! This makes debugging of conflicts between packages easier, as you have them all together.

2.3. macros.tex

This section introduces the use of the predefined macros. You are free to extend this list of macros in the file *template/macros.tex*.

2.3.1. todonotes

todonotes is a very useful package to track open items within a document and not mess around with special identifies to search for your TODOs.

List of TODOs

In the *thesis.tex* document, you can find the \listoftodos line. This generates a list of all TODOs defined in this document.

TODOs

In your document, you have two options to mark TODOs on your text. The first option is the one for short notes: \todonote. This todonote will pop up on the side of the document and point to the location in the paragraph, where it is linked to. Do not use if for long texts. In case you have long notes that you want to remember, you can use \todoinline.

Short TODO *()*

This is a very long TODO note that might go across multiple lines. You can really write some meaningful text in here and still read it later on. *()*

You will notice, that after such an inline TODO, the text is indented, even if you did not create the paragraph yourself. But as you are on a work-in-progress document, this does not matter too much.

REVIEW

REVIEW

The review notice shows that a section is ready for review. Use it to show your supervisor that you would like to get feedback for this section. Just add the *\review* command after the headline and the according information will be inserted into the PDF document.

FEEDBACK

FEEDBACK Is this a correct use of a feedback mark?

If you want to document a question you have for your supervisor, just use the command *\feedback*. This will generate an inline comment with the note that feedback is required.

2.3.2. Text Suggestions

We developed a macro to indicate suggestions for change. This macro keeps the old text striked through and prints the suggested text colored in olive next to it. For example, ~~this is the old text~~ this is the new text .

2.3.3. Marked Text

You do not need to create TODOs for all elements that you need to recheck. For example, you can just mark text sections with the command *\marktext* and the macros will highlight this text for you. These are not listed in the list of todos.

2.3.4. Thesis Goals and Numbered Elements

In the file *template/macros.tex*, we included the definition of goals for your thesis. You can use the numbering of goals through the section *goal* as used in Section 1.2. If necessary, you can also create such numbering for other elements that come up in your thesis. One student started the thesis by defining anomalies that should be detected with an Intrusion Detection System (IDS). He used this mechanism to reference the individual anomalies.

2.4. Acronyms

When writing your thesis, you might use a lot of acronyms. In order to have the fully written on the first use, and after that only use the short version you would need to go through the document at the end. Therefore, we recommend to use the following package: *glossaries-extra* with the module *acronym*. In the file *content/acronyms.tex*, we defined a few acronyms as example and will explain the use in the following.

In your text, you should always use the command *\gls{gPTP}* to call an acronym. With that, the first appearance will include both, short and long form (to visualize the effect, we mark the output red): Generalized Precision Time Protocol (gPTP) With the second use, you will only have the short form: gPTP

If you need to use the full term, you can use the command *\acrlong{gPTP}*: Generalized Precision Time Protocol. Alternatively, you can force the short version with the command *\acrshort{gPTP}*: gPTP.

To update the list of acronyms, you must enter the following command in your terminal: *makeglossaries thesis* Please remember to do that also before you generate your final version.

2.5. Finalize your Document

2.5.1. Remove unused Tables of Entries

In the *thesis.tex*, you find a number of tables for different kinds of entries (see \listof.....). Depending on if you use Tables or Code snippets, you need them, or you don't. Please remove all unused ones, such that you do not have empty ones in the final document. Also make sure, that the \listoftodos is empty and then remove it as well.

2.5.2. Preparing for the Print

If you want to print your document with a binding, you should adjust the document, depending on the type of print. Most likely, you will have a binding on the left. Therefore, you need to adjust the use of the package *geometry* in *template/packages.tex* to include the *bindingoffset=15mm*. Otherwise, you will squeeze you text into the binding on the left. Additionally, you need to choose, if you will print on both sides of the paper or only on the front side. For the front side only, you need to set the option *oneside* in the first row of the *thesis.tex* file. Make sure, that you configure the printer to only use the front side. Alternatively, you can tell teh document to start chapters on the right-hand side and also print on the back side of the paper. For this, you need to set the option *twoside* in the first row of the *thesis.tex* file instead of the *oneside*.

3. Related Work

The related work section is often underrated. Make sure that you focus on this topic BEFORE your thesis, but do not leave it out of sight during and after the thesis.

3.1. Structure

The related work section can have different structures:

Grouped by Topic The easiest way for you is to group the related work by topic. This means that you group the literature by common attributes and discuss them together.

Grouped by Relevance You can also group the work by relevance. This means, you start with work impacting your thesis the most. Towards the end of the section, you introduce related work least relevant.

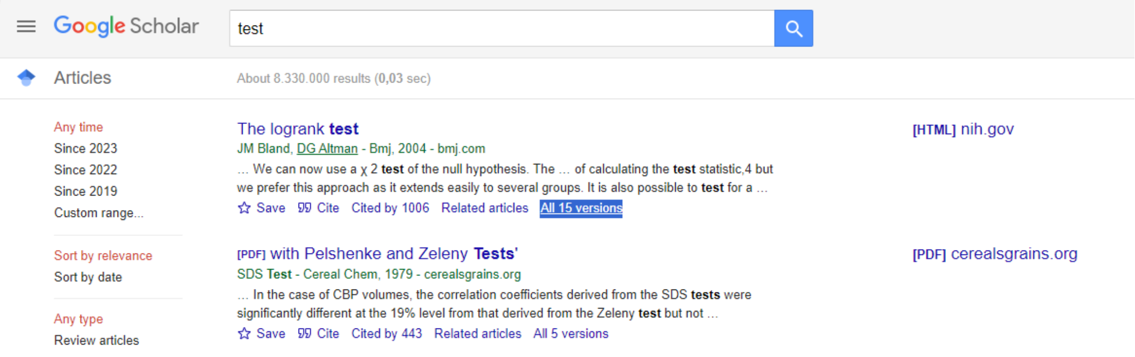
We will give a small example on the grouping by topic in the following.

3.2. Searching for Literature

A related work chapter starts with the literature research. There are two main sources, and many others, which you should use in parallel! In general, you should not only use the keywords that you come up with first, but continue your search with synonyms or other phrasings for the same topic.

Google Scholar [2] is the largest search engine for scientific literature. Sometimes, you do not find the PDF version of a document directly, but below the documents, you can find the *All XYZ versions* link, as shown in Figure 3.1. Often, this helps to get the PDF version without a paywall.

Another good source for scientific literature is ResearchGate [3]. Many authors upload the full text versions of their papers here. Alternatively, you can request the full text by the authors (sadly, they often do not respond...).



About 8.330.000 results (**0,03** sec)

The logrank **test**
JM Bland, DG Altman - Bmj, 2004 - bmj.com
... We can now use a χ 2 **test** of the null hypothesis. The ... of calculating the **test** statistic,4 but we prefer this approach as it extends easily to several groups. It is also possible to **test** for a ...
☆ Save 99 Cite Cited by 1006 Related articles All 15 versions
[HTML] nih.gov

[PDF] with Pelshenke and Zeleny **Tests'**
SDS **Test** - Cereal Chem, 1979 - cerealsgrains.org
... In the case of CBP volumes, the correlation coefficients derived from the SDS **tests** were significantly different at the 19% level from that derived from the Zeleny **test** but not ...
☆ Save 99 Cite Cited by 443 Related articles All 5 versions
[PDF] cerealsgrains.org

Figure 3.1.: An example search in Google Scholar

You need to enter all references that you want to use in the *content/thesis.bib* file. Both, Google Scholar and ResearchGate give you export formats in BibTex style, such that you can copy and paste the entry. Additionally, some tools presented in Section 3.4 do the same thing. However, please make sure to double-check the BibTeX entry yourself!! One common mistake is that you assume the capitalization stays the same, but it does not. Make sure to use double brackets {{*SOME TEXT*}} around the text you want to keep the capitalization. Finally, you can also use the library DBPL [4] to retrieve well maintained BibTeX entries for a very large number of scientific literature.

With scientific literature, it is important that you state the authors name, if you refer to their work in detail. For example, Müller et al. [5] present a scanning pipeline for the analysis of software versions on the internet. This pipeline includes banner grabbing to retrieve version information of services. It is important that you put the ˜ between the name (always use the first author's last name and add *et al.* if there are multiple authors) and the \cite. To have a good understanding of the differences to your work, you **need** to put a relation between your work and the related work. Compared to Müller et al., we also include the probes for cryptographic configurations. With that, we are able to analyze the landscape of offered cypher suites in, e.g., TLS or OPC UA.

Alternatively, you can also group literature that is very similar. The modeling of TSN networks is well researched for numerous use cases [6, 7], including detailed forwarding latency, jitter, and interference models. However, none of these components models load dependent delays, such as firewalls. Therefore, we present the first model of a firewall to be included in TSN network simulation. We recommend grouping literature only, if they are far away from the own research, and you want to show that a certain field of research (not too important for you) is well covered.

3.3. Writing Scientific Text

In general, scientific text differs from what you learned in school. The text should not be entertaining or exciting, but present technical information. Therefore, do not be afraid to repeat technical terms, without finding synonyms. Also, keep your sentences simple and short (especially in English).

3.3.1. In German

German texts are typically written in a passive form.

Add some information on German texts *()*

3.3.2. In English

Writing in English is different from German. In English texts, you use active voice. A good source for detailed information is the book "Scientific Writing" from Justin Zobel [8]. Students from HS Esslingen can read it for free.

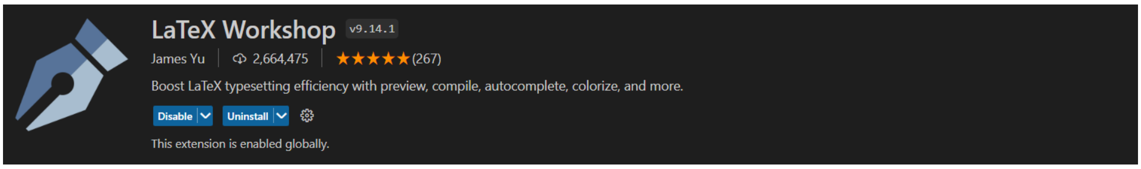
Add some information on English texts *()*

3.4. Tools to Support Writing

Include details on tool support *()*

3.4.1. Tools for Writing

A very useful tool for your thesis is Visual Studio Code (VS Code) [9]. With this tool, all programming, but also writing of the thesis is made easier! To write your thesis with



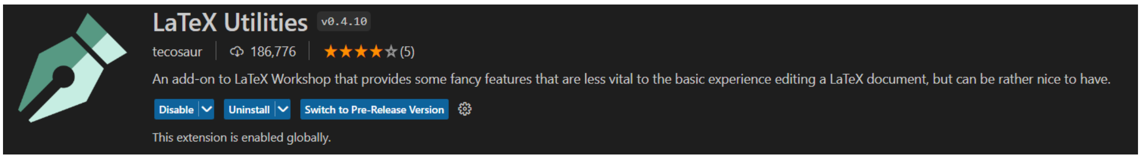
LaTeX Workshop v9.14.1

James Yu | 2,664,475 | ★★★★★ (267)

Boost LaTeX typesetting efficiency with preview, compile, autocomplete, colorize, and more.

This extension is enabled globally.

Figure 3.2.: VS Code Plugin: LaTeX Workshop



LaTeX Utilities v0.4.10

tecosaur | 186,776 | ★★★★☆ (5)

An add-on to LaTeX Workshop that provides some fancy features that are less vital to the basic experience editing a LaTeX document, but can be rather nice to have.

This extension is enabled globally.

Figure 3.3.: VS Code Plugin: LaTeX Utils

VS-Code, we recommend the following extensions:

- VS Code Extension: LaTeX Workshop (c.f. Figure 3.2)
- VS Code Extension: LaTeX Utils (c.f. Figure 3.3)
- VS Code Extension: LTex - LanguageTool (c.f. Figure 3.4)

With these extensions, VS Code will automatically compile the LaTeX code of this thesis template and the previewed PDFs are updated automatically in the background. You can view the compiled PDFs in VS-Code using the shortcut *cmd + alt + v*. Other useful shortcuts include:

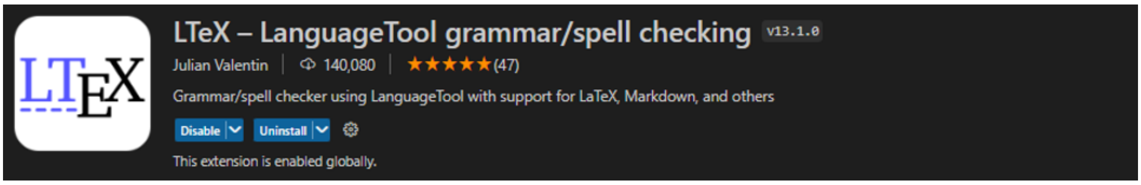
- jumping from the code to the PDF:
  - **mac:** *cmd + option + j*
  - **windows/linux:** *ctrl + alt + j*
- jumping from the PDF to the code:
  - **mac:** *cmd + click*
  - **windows/linux**: *ctrl + click*

Additionally, *LTex* will highlight grammar and spelling mistakes. Please follow the documentation of this plugin to change the language to something else than English if necessary.

3.4.2. Tools to Organize References

We reccomend the use of the tool Zotero [10] to manage your references.

Include details on Zotero *()*



LTeX – LanguageTool grammar/spell checking v13.1.0

Julian Valentin | 140,080 | ★★★★★ (47)

Grammar/spell checker using LanguageTool with support for LaTeX, Markdown, and others

This extension is enabled globally.

Figure 3.4.: VS Code Plugin: LTeX - LanguageTool

4. Ontology Design

4.1. Dataset Generation

Given the lack of publicly available open-source datasets specific to German cuisine, we opted to construct our own dataset through web scraping. Primary data sources included Wikipedia entries related to traditional German dishes and beverages, supplemented by additional culinary websites to ensure broader coverage and diversity. The initial dataset comprised solely the names of dishes and beverages, lacking the essential attributes and relationships required for constructing a meaningful Knowledge Graph. To address this limitation, a data augmentation strategy was implemented to enrich the dataset with additional semantic information.

4.1.1. Data Augmentation

To enrich the sparse initial dataset, a data augmentation strategy was employed using an open-source Large Language Model (LLM) via the Ollama framework. The LLM was prompted to act as a domain expert in German cuisine and beverages. It processed the dataset row by row and completed the remaining attributes based solely on the given `DishName` (for dishes) or `BeverageName` (for beverages). The goal was to semantically enrich each entry with structured information relevant for ontology construction and Knowledge Graph population. The columns to be completed for each dataset were as follows:

Dishes Dataset

`DishName`, `Description`, `Region`, `MainIngredient`, `Ingredient`, `StateOfMainIngredient`, `DietType`, `MealEatenAtPartOfDay`, `Variation`, `FlavorProfile`, `PreparationMethod`, `PreparationTimeMinutes`, `MeatCut`

Beverages Dataset

`BeverageName`, `Description`, `Region`, `MainIngredient`, `Ingredient`, `FlavorProfile`, `IsCarbonated`, `AlcoholContent`, `BeverageType`, `ServingTemperature`, `IsGermanStaple`

Models Used

The models employed and evaluated for data augmentation include:

- **Mistral** – 7 billion parameters, 32K context window, developed by Mistral AI
- **Gemma 3** – 12 billion parameters, 128K context window, developed by Google
- **Llama 3** – 8 billion parameters, 8K context window, developed by Meta

To determine which model produced the highest quality dataset, the top 5 rows from each augmented dataset were manually inspected. Leveraging domain knowledge, a decision was made regarding which dataset best suited the objectives of Knowledge Graph construction. Below a row excerpt is depicted and the various LLM response styles are shown.

4.1.2. Mistral:7b - Dataset Excerpt

Attribute	**Value**
Dish Name	Apfelkren
Description	Apfelkren is a classic German condiment made from grated apples and horseradish.
Region	Thuringia
Main Ingredient	Apples, Horseradish
Ingredients	Grated Apples, Fresh Horseradish Root, Salt, Vinegar
State of Main Ingredient	Grate apples, grated horseradish
Diet Type	omnivore
Meal Eaten At Part of Day	anytime
Variations	Sweetened versions with sugar or honey also exist
Flavor Profiles	tangy, mild
Preparation Method	Grate apples and horseradish, season with salt and vinegar
Preparation Time (Minutes)	15
Meat Cut	—

Table 4.1.: Apfelkren Dish Details - Mistral generated

4.1.3. Llama3:8b - Dataset Excerpt

Attribute	**Value**
Dish Name	Apfelkren
Description	A sweet and tangy condiment made with apples
Region	Bavaria
Main Ingredient	Apples
Ingredients	onions, mustard seeds, vinegar
State of Main Ingredient	Raw
Diet Type	Vegetarian
Meal Eaten At Part of Day	Anytime
Variations	—
Flavor Profiles	Sweet, Tangy
Preparation Method	Chopping and mixing
Preparation Time (Minutes)	15
Meat Cut	—

Table 4.2.: Apfelkren Dish Details - Llama3 generated

4.1.4. Gemma3:12b - Dataset Excerpt

Attribute	**Value**
Dish Name	Apfelkren
Description	Mashed apples with horseradish, often served as a condiment
Region	Bavaria, Swabia
Main Ingredient	Apple
Ingredients	apple, horseradish, lemon juice, salt, pepper
State of Main Ingredient	mashed
Diet Type	vegetarian, vegan, omnivore
Meal Eaten At Part of Day	Lunch, dinner
Variations	with vinegar, with sugar
Flavor Profiles	sour, fruity, aromatic, tangy
Preparation Method	mashed
Preparation Time (Minutes)	10
Meat Cut	—

Table 4.3.: Apfelkren Dish Details - Gemma3 generated

4.1.5. Data Augmentation – Evaluation

In the case of the instance *Apfelkren*, all evaluated models successfully identified the dish as a condiment in the *Description* column. However, with regard to regional attribution, the **Gemma 3 (12B)** model demonstrated the highest degree of precision. Particularly noteworthy is Gemma 3's performance in assigning dietary classifications: while other models limited the *DietType* to a single label, Gemma 3 provided a more inclusive categorization by assigning multiple applicable labels — namely, *vegetarian*, *vegan*, and *omnivore* — thus better reflecting the dietary flexibility of the dish. Across the full dataset, Gemma 3 consistently outperformed the other models in terms of:

- lexical and semantic precision,
- descriptiveness of the generated content,
- nuanced handling of dietary classifications,
- and regional attribution accuracy.

Based on these observations, the **Gemma 3 (12B)** model was selected as the primary model for data augmentation. Its outputs will serve as the foundation for generating instances in the ontology-driven Knowledge Graph construction. Furthermore, Gemma 3 will also be applied to the German beverages dataset, under the reasoned assumption that its superior performance in the dish dataset is likely to generalize adequately in the domain of German beverages.

Table 4.4.: Example transmission delays.

Size	100 Mbit/s	1 Gbit/s	2.5 Gbit/s	10 Gbit/s
64 B	6.7 µs	672 ns	269 ns	67 ns
123 B	11.8 µs	1.2 µs	470 ns	118 ns
1,522 B	123.4 µs	12.3 µs	4.9 µs	1.2 µs

The structure for the design chapter strongly depends on the topic of your thesis. Throughout the process, you will refine this structure with your supervisor.

If you have a mathematical content in your thesis, you can use equations as follows. Each equation uses its own space and should not be longer than one line, otherwise it is difficult to read. For this document, the content of the equation is not important. Still, we reference and explain it as an example. Equation 4.1 defines the set of streams $\mathcal{S}_{\text{ifr}}^{e,s}$that will interfere with stream s if no Time-Aware Shaper (TAS) is configured. For the purpose of consistent formulas, we introduce commands for complex terms in the file *content/definitions.tex*

$$\mathcal{S}_{\text{ifr}}^{e,s} = \{g | g \in \mathcal{S}^e \wedge g \neq s \wedge p_g \geq p_s\} \tag{4.1}$$

In your thesis, you can also use more complex equation structures. This structure still has one equation per line, but aligns them to be grouped together and indent the same way. The alignment happens through the & across the lines. This complex structure still has references to each of the individual formulas. Therefore, we can easily explain each of the Equations 4.2, 4.3, and 4.4 in the text.

$$\text{case C1: } t_{\text{enqCT}}^{e,s} < t_{\text{open}}^{e,s} \tag{4.2}$$

$$\text{case C2: } t_{\text{enqCT}}^{e,s} + d_{\text{dwell}}^{e,s} + d_{\text{ifrcross}}^{e,s} < t_{\text{close}}^{e,s} \tag{4.3}$$

$$d_{\text{gate}}^{e,s} = \begin{cases} t_{\text{open}}^{e,s} - t_{\text{enqCT}}^{e,s} + d_{\text{ifrpath}}^{e,s} & \text{case C1} \\ 0 & \text{case C2} \\ CT_{\text{GCL}}^{e,s} - t_{\text{enqCT}}^{e,s} + t_{\text{open}}^{e,s} + d_{\text{ifrpath}}^{e,s} & \text{otherwise} \end{cases} \tag{4.4}$$

For your thesis, we recommend the use of tables to summarize series of data. Sometimes, tables are easier to read than a lot of text with data. In Table 4.4, we present the transmission delay of frames with different sizes at different link speeds. Instead of writing all these numbers in a long paragraph, we can now use the time to highlight key insights. For example, you can see that a frame with 1,522 B only consumes about twice the time at 1 Gbit/s, compared to a frame with size 64 B at 100 Mbit/s.

In your thesis, you might want to explain your solution with Pseudocode algorithms. For example, Algorithm 4.1 presents a sorting algorithm for firewall rules. In Line 4, we compare if a rule depends on the current group.

Finally, you can also use theorems, and lemmas in your thesis. Please make sure (as always) to reference each of them in the text and explain their meaning. For example, Theorem 4.1, or Lemma 4.2.

Theorem 4.1. *Let f be a function whose derivative exists in every point, then f is a continuous function.*

Algorithm 4.1 Group Sort

```
for $x \in RuleSet$ do
    $group \leftarrow [x]$
    for $y \in RuleSet[x.idx:]$ do
        if $dependency(group, y)$ then                    ▷ Extend group
            $group \leftarrow group\#y$
        else
            if $hit\_count([y]) < hit\_count(group)$ then
                                                          ▷ Move complete group
                $y.idx \leftarrow x.idx$
                for $z \in group$ do
                    $z.idx \leftarrow z.idx + 1$
                end for
            end if
            $group \leftarrow [y]$
        end if
    end for
end for
```

Lemma 4.2. *Given two line segments whose lengths are a and b respectively there is a real number r such that $b = ra$.*

For formal procedures and algorithms, use the unified notation of proofs to state your conclusions. An example for that follows here:

Proof. To prove it by contradiction try and assume that the statement is false, proceed from there and at some point you will arrive to a contradiction. □

5. Knowledge Graph Construction

knowledge graph

6. Chatbot Architecture

In your implementation section, you can also use code listings to explain the details of your work. To include the code you have two options: A) use the inline method for short one liner: `[elem for elem in my\_list if elem.variable==filter_value]`, or B) use multiline listings to visualize more complex examples. Similar with figures, make sure to always reference your multiline listings and explain what they do. In Listing 6.1, you find the function `my_filter` with two parameters. This function uses the list in the first parameter for all elements with the variable equal to the second parameter.

```python
def my_filter(my_list, filter_value):
    new_list = []
    for elem in my_list:
        if elem.variable == filter_value:
            new_list.append(elem)
    return new_list
```

Listing 6.1: A filter function in Python

If you have even more detailed code listings, they belong into the appendix. For example, you can find a filter function with an additional sorting method in the Appendix A.1.

In some cases, it might happen that your inline code does not fit into the current line anymore. For example with this code here: `[elem for elem in my\_list if elem.variable ↪ ==filter_value]` But don't worry, the arrow in the beginning of the line will highlight that!

If you require further details on the use of listings in LaTeX, you can find additional information on listings at the following two locations:

1. https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings

2. https://www.overleaf.com/learn/latex/Code_listing

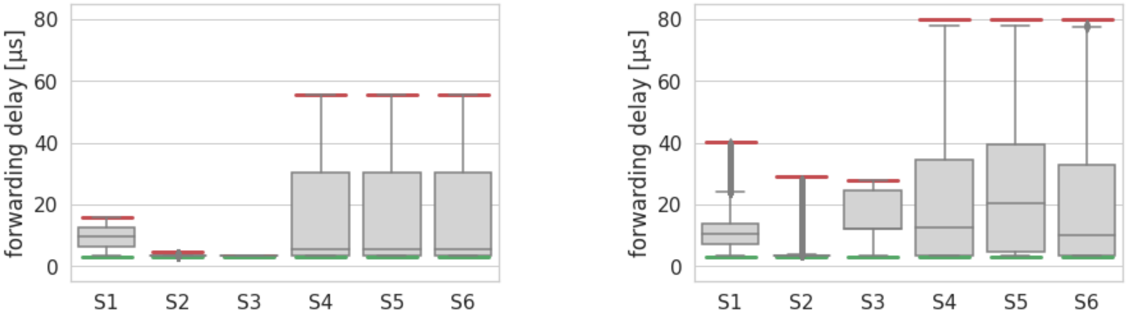
7. Evaluation

The content, e.g., measurements, survey, or simulations, of the evaluation depends on your thesis. What will not depend on your thesis that you have results that you will visualize in figures. In general, use one figure after each other and explain how to read it and directly discuss the content and **meaning**. Sometimes, you want to visualize a change through your design and implementation, or want to compare two systems. Then, you should place the figures either above each other or next to each other. The LaTeX package *subcaption* helps with that, and it looks as follows.

Figure 7.1 shows two figures with different configurations during the measurement. Table 7.1 presents six different settings for the evaluation of a single node. We use these settings throughout the evaluation with the labels S1 to S6. Figure 7.1a shows the measured forwarding delay for the six different settings without interference on a single node. Each measurement contains at least 1,000 packets and is visualized as box plot. Figure 7.1b shows measurements for the same six settings, but this time with interference by other traffic. One can clearly see that the majority of the results (the box represents 50% of all measurements) is similarly distributed for the settings S4 to S6. However, we also see a lot of outliers, raising the worst-case latency to 80 μs. For the settings S1 to S3, we see the same outliers, caused by the interference. Across all settings, we observe the settings S2 and S3 having the best performance, without and with interference.

Table 7.1.: Single Node Evaluation Settings

Setting	**Description**
S1	Strict Priority
S2	FP with priority 7 in the express category
S3	synchronized TAS with prio. 7 in the TAS window
S4	unsynchronized TAS with prio. 7; CT^s_{app} = 100µs
S5	unsynchronized TAS with prio. 7; CT^s_{app} = 45µs
S6	unsynchronized TAS with prio. 7; CT^s_{app} = 196µs



(a) no interference

(b) interference

Figure 7.1.: Single node queuing delay evaluation

8. Conclusion

The conclusion is the last chapter of your thesis. Make sure to match its content to your design, implementation, and evaluation. Also, that you recap your introduction and motivation to ensure you solved the actual problem.

8.1. Summary

The summary recaps the complete thesis. Make sure to highlight the most important pieces of your work with a reference to the introduction. Additionally, present the applicability of your design and implementation.

8.2. Future Work

Which tasks, problems, or ideas are still open? In the future work section, you should highlight these open points and give reasons for their relevance.

9. Bibliography

[1] D. Evans, P. Gruba, and J. Zobel, *How to Write a Better Thesis.* Springer, 2014. [Online]. Available: https://doi.org/10.1007/978-3-319-04286-2

[2] Google, "Google Scholar," 2023. [Online]. Available: https://scholar.google.com/

[3] ResearchGate GmbH, "Research Gate," 2023. [Online]. Available: https://www.researchgate.net/

[4] Schloss Dagstuhl - Leibniz Center for Informatics, "International Workshop/Symposium on Database Programming Languages (DBPL)," 2023. [Online]. Available: https://dblp.org/db/conf/dbpl/index.html

[5] R. Müller, J. Ruppert, K. Will, L. Wüsteney, and T. Heer, "Analyzing the software patch discipline across different industries and countries," in *Sicherheit, Schutz und Zuverlässigkeit: Konferenzband der 11. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), Sicherheit 2022, Karlsruhe, Germany, April 5-8, 2022*, ser. LNI, C. Wressnegger, D. Reinhardt, T. Barber, B. C. Witt, D. Arp, and Z. Á. Mann, Eds., vol. P-323. Gesellschaft für Informatik, Bonn, 2022, pp. 159–170. [Online]. Available: https://doi.org/10.18420/sicherheit2022_10

[6] D. Hellmanns, L. Haug, M. Hildebrand, F. Dürr, S. Kehrer, and R. Hummen, "How to optimize joint routing and scheduling models for TSN using integer linear programming," in *RTNS'2021: 29th International Conference on Real-Time Networks and Systems, Nantes, France, April 7-9, 2021*, A. Queudet, I. Bate, and G. Lipari, Eds. ACM, 2021, pp. 100–111. [Online]. Available: https://doi.org/10.1145/3453417.3453421

[7] L. Wüsteney, D. Hellmanns, M. Schramm, L. Osswald, R. Hummen, M. Menth, and T. Heer, "Analyzing and modeling the latency and jitter behavior of mixed industrial TSN and detnet networks," in *Proceedings of the 18th International Conference on emerging Networking EXperiments and Technologies, CoNEXT 2022, Roma, Italy, December 6-9, 2022*, G. Bianchi and A. Mei, Eds. ACM, 2022, pp. 91–109. [Online]. Available: https://doi.org/10.1145/3555050.3569138

[8] J. Zobel, *Writing for Computer Science.* Springer, 2014. [Online]. Available: https://doi.org/10.1007/978-1-4471-6639-9

[9] Microsoft, "Visual Studio Code," 2023. [Online]. Available: https://code.visualstudio.com/

[10] Corporation for Digital Scholarship, "zotero," 2023. [Online]. Available: https://www.zotero.org/

A. Appendix

A.1. A Filter and Sort Function in Python

```python
def my_filter_with_sort(my_list, filter_value, reverse=False):
    new_list = []
    for elem in my_list:
        if elem.variable_a == filter_value:
            new_list.append(elem)
    new_list.sort(key=lambda x : x.variable_b, reverse=reverse)
    return new_list
```

Listing A.1: A filter and sort function in Python

Even in the appendix, never just paste code without referencing it from the text. Listing A.1 presents a function that filters a list for all elements containing a specific value in *variable_a*. Finally, the function sorts the remaining elements based on the variable *variable_b*. With the parameter *reverse*, you can specify if the sorting should be ascending or descending.