POLITECNICO DI MILANO
Master of Science in Computer Science and Engineering
Dipartimento di Elettronica, Informazione e Bioingegneria

[Title of the Thesis]

Supervisor: [Name]
Co-supervisor: [Name]

M.Sc. Thesis by:
[Name], matriculation number [nnn]
[Name], matriculation number [mmm]

Academic Year 200N-200N+1

About this template

With this template I want to give you some input on how to structure your thesis if you develop your thesis with me in Politecnico di Milano. Next to the pure structure, which you should reuse and adapt to your own needs, the document also contains instructions on how to approach the different sections, the writing and, sometimes, even the work on your thesis project itself. Sometimes you will also find boxes like this one. These are meant to provide you with explanations and insights or hints that go beyond the mere structure of a thesis.

I hope this template will help you do the best thesis ever, if not in the World, at least in your life.

Florian Daniel
October 12, 2017

Disclaimer: Sometimes I may make statements that are general, if not over-generalized, personal considerations, or give hints on how to do work or research. Be aware that these are just my own opinions and by no way represent official statements by Politecnico di Milano or its community of professors. If something goes wrong with your thesis or presentation, you cannot refer to these statements as a defense. You are the final responsible of what goes into your thesis and what not.

Acknowledgements: The original template for this document was not created by me. I would love to acknowledge the real creator, but I actually do not know who it is. The template has been passed on to me by a former student, who also didn't know the exact origin of it. It was circulating among students. However, to the best of my knowledge at the time of writing, it seems that Marco D. Santambrogio and Matto Matteucci may have contributed at some point with considerations on structure and funny citations. Both were helpful and enjoyable when preparing this version of the template. I will be glad to add more precise acknowledgements if properly informed about the origins of this template.

Supervisors and co-supervisors

If the supervisor is internal to Politecnico di Milano (a professor or researcher), then on the first page use "Supervisor" plus the titles "Prof." and "Dr." for professors and researches, respectively. If the work was co-supervised by someone else, refer to him/her as the "Co-supervisor." If the work was supervised by someone external to Politecnico di Milano, use "External supervisor" for the external supervisor plus "Internal supervisor" for the internal supervisor that mandatorily must co-supervise the work with the external supervisor.

Optionally, here goes the dedication.

Abstract

The abstract is a small summary of the thesis. It tells the reader in few words (up to one/one and a half page of total text) everything he/she needs to understand:

- ☐ the *context* of the work (e.g., chatbots),
- ☐ the specific *problem* approached by the thesis (e.g., the development of personal bots by non-programmers),
- ☐ if applicable, clearly state the *research questions* you would like to answer (e.g., "is it possible to enable non-programmers to do X using A?"),
- ☐ the three/four *core aspects of the proposed solution* (e.g., use predefined rules, use machine learning, assisted development, etc.),
- ☐ the *concrete outputs* produced by the thesis (e.g., a state of the art analysis, a conceptual/mathematical model, an application, middleware or API, an empirical study with/without users, etc.), and
- ☐ the *findings and conclusions* that one can draw from the evaluation of the approach (e.g., that under some very specific conditions non-programmers are indeed able to implement own chatbots effectively using the proposed technique).

Checklists
Now and there I propose checklists with items, such as the one just above this box. They are meant for you to check if you included all the content that is relevant and that should be included, in order to make your text complete. When reading your thesis, I will look for all these items.

Writing style

This is a M.Sc. thesis. It's neither Facebook nor Twitter nor an email. This is going to be an official document with legal value that will decide on the final mark of your yearlong university career and perhaps even on your future work perspectives. So, you surely don't want to be judged badly because of grammar errors, flawed/wrong vocabulary or superficial layout and/or text structure. It is a must that what you write is always *correct* content- and language-wise (no false statements or claims, no language mistakes), *readable* (no sentences that cannot be understood) and targeted at the *average-skilled reader* (professors, but also your own colleagues).

Plagiarism

This is a M.Sc. thesis. It's neither Facebook nor Twitter nor an email. This is going to be an official document with legal value that will decide on the final mark of your yearlong university career and perhaps even on your future work perspectives – yes, I plagiarized myself here a little bit. So, you surely don't want to copy/paste material from scientific articles, online resources, books, and similar without adequately acknowledging the holders of the respective intellectual property rights. If you do so, it is a must that you properly *cite* each source where you take text or inspiration from. It is fine to do so – actually, citing someone is a compliment! – but it becomes a crime if the source is not cited. Not only M.Sc. titles but also Ph.D. titles have been withdrawn for fraudulent "reuse" of others' intellectual property. Be aware that Politecnico di Milano, like most higher educational institutions that issue university degrees or scientific publishers, may use specialized software to automatically detect plagiarism.

Sommario

Here goes the translation into Italian of the abstract. If the thesis is written in Italian, no translation into English is needed. Hence, one of the following must be checked:

☐ Thesis written in *English*, properly proofread translation needed

☐ Thesis written in *Italian*, no translation needed, chapter omitted

Acknowledgements

If you would like to thank somebody for given support, this is the right place to do so.

Contents

Chapter 1

Introduction

The introduction is one of the core chapters of your thesis. It expands what has already been said in the abstract with additional details on the content and contribution and on the structure of the thesis. It is meant to introduce the reader to the work he/she will be reading in the rest of the document and, most importantly, to get the reader curious about reading on, knowing more about your work.

1.1 Context: [topic]

This thesis is about describing the work you are doing in your final thesis project. You have been working on it for months, and nobody knows the work better than you do. This is great and exactly how things should be: by doing your thesis project you became an expert – if not *the* expert – in this specific field of research and/or technology.

But attention: being the expert is also dangerous when it comes to explaining others what you did and why you think you did a great work that deserves attention (I give it for granted that you work does so). There are only very few people around you (your supervisor and possible co-supervisor, some friends, maybe someone else) who are as expert as you are in this topic. So, if you start in a full-impact fashion to tell that you implemented an extraordinarily cool, new algorithm to solve X, or that you discovered this extremely surprising finding Y, or that you mathematically proofed that Z, etc. (you got it), your reader will not understand anything. Therefore, before talking about what you actually did, you need to introduce the reader to the context of your work, provide the necessary core definitions that are needed to understand the terminology you will be using in the rest of the thesis (if it's not standard IT terminology).

Therefore:

- ☐ Tell the *research area(s)* your work/project focuses on. If you are doing your thesis with me, likely candidates of research areas are Web Engineering, Data Science, Crowdsourcing, Service-Oriented Computing, Business Process Management.
- ☐ Tell possible *sub-areas* that are more specifically related to what you are doing. Again, if you are doing your thesis with me, likely candidates of sub-areas are chatbots, social knowledge extraction, business process matching/modeling, quality control in crowdsourcing, etc.
- ☐ Make the *heading* of your context section self-explaining by substituting "[topic]" in heading 1.1 with the sub-area most relevant to your work. It should read like "Context: quality control in crowdsourcing" or similar.
- ☐ If needed, introduce some *key definitions* (no need to introduce everything here, but be sure that the introduction does not use terminology the reader may not be familiar with). For instance, if you are working on chatbots, this is definitely a term that needs to be introduced here; it's not yet commonly known but it's crucial for the understanding of the rest of the thesis and introduction.
- ☐ Use *examples* to make definitions and ideas concrete and clear.
- ☐ Throughout, make *references* to the relevant literature.

Use of tenses and pronouns

Writing a thesis is writing a scientific document like scientific articles or research publications. There are two conventions that are usually applied in this kind of publications (admittedly, they may seem somewhat odd if not used to):

First, the most used tense is the *simple present*. The thesis is meant to describe a piece of work, from problem statement, to the conception of a solution, its implementation and evaluation. Yet, it's not a novel about your life, and it's not meant to provide a chronological story about what you did and didn't do. Content is presented in an order that is most effective to convey its message, not in time order. In this spirit, it's much more effective to say "in order to get result A, first we do X, then we do Y and then Z," instead of saying "in order to get result A, we did Y after having done X, then we went on doing Z." The order of actions, their interconnections, inputs and outputs already tell the dependency – if properly described. Most of the times, the most effective way to describe a solution or

methodology only becomes clear after trial and error. It's enough to explain the result, not how you got there chronologicallly.

Second, the *pronoun* used to talk about the own work is "our" (work). That is, it is custom to say "we" instead of "I," even if you are writing your thesis alone. However, don't forget about all the people that helped you get there: your supervisor, co-supervisor, colleagues, etc. This may sound strange at the beginning, but, at the other hand, using "I" too often risks to convey the impression that you are self-focused and egoistic, which is never good.

1.2 Scenario and Problem Statement

Now that the reader got the general context of your work and has an intuition of the problem you will be solving in the rest of the thesis, it's time to be clear about which *specific problems* your thesis project is going to solve. One way of doing so is by describing a *scenario* (a description of a real situation, with all its actors, roles, tasks, instruments, etc.) that provides evidence that there are one or more real problems right now that, with the current technology and understanding of the domain, are hard to solve or not solvable at all. If instead the problem(s) can be solved already, it should be evident from the scenario that this is possible only at a prohibiting cost or with unsatisfying guarantees on the quality of the result or not within useful time for the target user.

It's important that the scenario is written in such a way that the reader, after reading it, agrees with you that the problem you are focusing on is a relevant one, one that deserves being studied and solved. Consider that if you convince the reader here that your thesis is needed (after all, that's what this section is about), he/she will be very open to possible solutions and happy to see how you solve it. If instead you fail to convince the reader – let me be harsh – the whole rest of your thesis is useless in the eyes of the reader. This is the worst outcome you want.

Conclude this section by explicitly stating which of the problems evident in the scenario you are approaching. Don't raise false expectations! Never ever tell the reader there are five core problems and then solve only two of them in the thesis, without telling upfront that this is what you intended to do in the first place. As soon as you list problems, the reader wants to see a solution, unless you stop him/her immediately from thinking so by telling that out of the described problems you focus on a subset only, usually because this subset is already a huge research and development problem in its own.

In summary:

- ☐ Describe a *real scenario* that provides evidence of *real problems*.
- ☐ Convince the *reader* that the problems need to be solved.
- ☐ Use an *illustration* or *figure* to help the reader understand.
- ☐ If possible, provide *references* to literature that backs your assessment of the problem.
- ☐ Provide a clear *problem statement* that summarizes what came out of the scenario and your specific focus.

1.3 Methodology

Fixed the problem(s) you want to approach, you can approach it/them in thousands of different ways. Your way is just one of the thousands, and the reader may have (and very likely will have) a very different intuition of how to solve the problem(s) you just pointed out. So, clarify how you intend to proceed:

- ☐ Tell if you follow an existing *methodology* or not; if yes, name it and provide a reference to literature, if available. For example, Design Science [1] is a likely methodology to cite here.
- ☐ Tell which of the following *procedures*, *techniques*, *methods* you use in your work and for which purpose (put them also into the right order, so that their application or use makes immediate sense to the reader):
  - ☐ *Systematic literature review, survey*
  - ☐ *Statistical hypothesis formulation and testing*
  - ☐ *Software prototyping*
  - ☐ *Iterative development*
  - ☐ *Participatory design*
  - ☐ *Performance evaluation*
  - ☐ *Comparative studies*
  - ☐ *User studies*
  - ☐ *Expert interviews*
  - ☐ *Simulation/emulation*

- ☐ *Live experiments*
- ☐ *Case studies*
- ☐ *Mathematical theorem proofing*
- ☐ *Mathematical modeling*
- ☐ *Pseudocode*
- ☐ *Graphical modeling* (e.g., UML, ER)
- ☐ *Model-driven development*
- ☐ *Automatic code generation*
- ☐ ...

☐ Tell if you use some special *software instruments* that help you in your work. We are of course not talking about Word or Google Search. Perhaps you can tell that you used R for data analysis or some specific modeling instrument for automated code generation or simulation.

1.4 Contributions

Now that the reader knows what you want to solve and how you intend to proceed, you can anticipate the contributions your thesis makes to the state of the art. Attention, a thesis project may produce lots of different *outputs* (e.g., a software prototype, a set of registrations and transcripts of interviews, datasets collected during experiments) and *contributions* (e.g., a demonstration that some software solutions solves a given problem under well defined conditions, a formal proof that some property holds, empirical evidence that something works as expected). The former are all the artifacts produced throughout the work. The latter refer to *new knowledge* (if you are doing a full thesis) or the most important, *final output* (if you are doing a tesina). Sometimes, outputs and contributions overlap, but not necessarily.

Typical contributions are (multiple choices may apply to your thesis):

☐ A *systematic literature review* of the state of the art providing evidence for some argument

☐ The design of a *model* (mathematical, graphical, algebraic, etc.) describing how to solve a real world problem in a reusable fashion

☐ The drawing of *conclusions* (findings) from the analysis of a dataset describing some physical or virtual phenomenon

☐ The implementation of a *software prototype* solving a real world application problem

☐ The design of a *language* (textual, graphical) enabling others to solve own problems or to solve them easier

☐ *Formal proofs* of correctness, completeness or other properties of the proposed models or theorems

☐ *Objective evidence* from empirical studies (e.g., performance analyses or simluations) that demonstrate that the proposed prototype or solution works / works better than existing software or solutions that solve the same/similar problem(s)

☐ *Subjective evidence* from user studies or expert interviews backing the claims of viability of the proposed problem or solution/artifact

☐ A reasoned *argumentation*, e.g., based on a detailed case study, supporting the viability of the proposed problem or solution/artifact

Thesis vs. Tesina

Let me spend some words on the difference between these two. Before that, however, it is important to clarify the very purpose of your final project, be it a thesis or a tesina (a small thesis). The purpose of it is giving you the possibility to show that, after years of attending classes and giving exams, you are also able to *apply* the knowledge you acquired during your studies. In short, it's all about you showing that you are *mature*. Mature form a knowledge perspective, mature from an application perspective, mature from a work/teamwork perspective, mature from an ethical perspective.

It is common that a thesis project is not very well defined in its beginning and that even the supervisor does not really know how to approach a given problem or which problem to focus on in the first place. This may even be annoying to you, but attention: there is no intention behind it. Your supervisor is not withholding information from you to test you or to see if you get something. It's just the nature of real *problem solving*. If things were clear from the beginning, there wouldn't be any problem! Fledging out the problem and agreeing on a solution and methodology is a core part of you demonstrating your maturity – if not the most important one. *How* you proceed from the inception of the thesis idea to the final solution is as important as *what* you find and/or produce in the end.

This being said, a *thesis* in Politecnico di Milano usually requires you to make a contribution to the literature (the so-called state of the art). Making a contribution – from a science point of view – means creating new *knowledge*, that is, finding something that nobody knew before, demonstrating a property that nobody showed

before, improving the performance of a given system with a new algorithm, and similar. For a thesis, it is therefore not enough to produce a perfectly engineered solution. It is key that you also demonstrate, provide empirical evidence or proof that your solutions performs as claimed. Well, for a *tesina* this last demonstration is usually not required, and the focus is on the engineering of the solution. In addition, perhaps in the case of the tesina the solution to be engineered is also less complex then for a thesis, but this depends on the context and on how you want to measure complexity.

1.5 Structure of Thesis

Here you explain the structure of the thesis, so that the reader knows how to read it. Consider that not every reader wants to read through the whole thesis to find some specific information. Actually, only few will do so (your supervisor and co-supervisor, and the possible reviewer for sure). Many more will just leaf through it and look for specific types of information (e.g., the context of your work, your findings, how you implemented something, which technologies you used). It is your duty to accommodate them all. How? By telling them how your thesis is structured.

Therefore, in this section you provide a brief description (2-3 sentences) for *each* chapter that follows this introduction. Use an itemized or numbered list to structure the text, like this:

- ☐ Chapter 2 introduces the state of the art and...
- ☐ Chapter 3 provides...
- ☐ ...

Structuring text
Besides telling the reader how the content of your thesis is organized into chapters, it is important that you master some basic text structuring techniques. To organize your text there are lots of instruments you can use: chapters, sections, sub-sections, paragraphs, itemized lists, numbered lists, code examples, figures, images, screen shots, captions below figures, tables, and so on. Use them all! Don't write text without structure. Never.

Be aware that the structure of your text, that is, how you present your work, conveys a lot of information about how well you actually understand what you are writing about, how much you care about being clear and helping your reader understand, and how much value you give yourself to your own thesis. A well

structured presentation of content that the reader can understand and agree with is a huge plus in this respect. Text that lacks proper paragraphs, does not use lists where needed, etc. is a minus and also much harder to read (think about how much a well structured text can help you go back ten pages and find concepts you know you read about compared to a text that comes without an easy to memorize formatting and structure). When writing, think about some of your textbooks. Since you are doing an engineering degree, I'm sure these are textbooks that make exemplary use of the different formatting instruments available.

Chapter 2

State of the Art

This chapter discusses the state of the art that is relevant for your own work. What does that mean? It means that it provides the reader with all the relevant references he/she may need to know in order to understand better three things: (i) the context of your work, (ii) the problem and the need for a solution, and (iii) the value of your contribution. You achieve this by citing works or scientific papers that solved the same or similar problems in the past. Citing does not just mean adding a references to the bibliography and printing a number here; it means you tell the reader about the merits and possible demerits of each of the references you feel relevant. Of course, doing so requires you to first read each reference and, most importantly, to understand it. There should be lots of references in this chapter.

It is advisable that you structure the chapter into sections in function of the topics you treat. If you do so, before starting with the first section of the chapter, explain the reader how you structure your discussion in one paragraph.

- ☐ *Read* relevant literature and or *test* related software or tools.
- ☐ *Summarize* your reading.
- ☐ Provide correct *references* (the bibliography in the end of this document).

2.1 [Topic one]

...

2.2 [Topic two]

...

2.3 Summary

Close the state of the art chapter with some words that connect the discussion of the references to your thesis. Pay attention that the reader understands why you discussed the works/topics you discussed and how they are related to what you do.

- ☐ Show that in the state of the art the *problem* you want to solve has not yet been solved or not been solved in an as efficient / effective / easy to use / cost-saving fashion as you target with your work.
- ☐ If your work has similarities with some *specific references*, point them out here and explain why these are particularly important to you. Perhaps you started your investigation from the outputs of a specific paper or you want to improve the performance of an algorithm studied earlier; it's good to mention this here.
- ☐ Attention: this is not yet the place where to anticipate *your solution*. You may give hints, but it's too early to make a comparison between your work and the state of the art, as the reader does not yet know anything about your work. This discussion can go into the final chapter.

Chapter 3

[Core contribution]: Goals and Requirements

This chapter splits the problem that so far was still at a relatively abstract and intuitive level of understanding down into fine-grained sub/problems, which then lead to concrete action items to be approached throughout the thesis project. This is the chapter where you show your understanding of the *problem*. As such, it is important, on the one hand, to show your competence and, on the other hand, to explain the reader what exactly you are going to work on.

- ☐ Replace the "[Core contribution]" in the title of the chapter with the name of the core contribution of your thesis work. If, for example, your contribution is the design and evaluation of a modeling language for the modeling of crowdsourcing processes, you could use something like "Modeling Crowdsourcing Processes: Goals and Requirements."

3.1 Concepts

In the introduction, you already introduced the core terminology needed to understand the preliminary problem statement. Here you may want to provide more details and more terminology, as things now get more concrete and new concepts may be needed to explain what you are working on.

- ☐ Provide all the *definitions* of concepts that you need to explain your work and that you did not yet introduce in the introduction.
- ☐ For each new definition, don't forget to provide clear *examples*.

3.2 Goals and Requirements

Here you repeat the initial problem statement of Section 1.2 and possibly refine it using the refined terminology introduced just now. Solving the problem is the goal of your thesis. Clarify who you think is the target user or beneficiary of your work. Then reason about the goals, considering the context of your work, your competences, possible constraints imposed to the potential solution, etc. and identify a set of *requirements* that you want to meet with your solution (by now, you should know about requirements from Software Engineering or other classes):

- ☐ *Functional requirements* (expected functionalities supported by the solution)
- ☐ *Generic non-functional requirements* (expected performance/quality levels)
- ☐ *Architectural requirements* (e.g., if your solution is to be integrated into an existing system)
- ☐ *Technological requirements* (e.g., if your solution must use given technologies)

Try to be concrete and not too abstract. After this section, the reader should really understand what to expect from your thesis. Ideally, you (or the reader) should be able to use the list of identified requirements as a checklist to be checked in the end of this document and, again ideally, for each requirement it should be possible to decide (true/false) if it is met or not. This may ask for the definition of suitable metrics to measure satisfaction. However, here it's too early to talk about that; this will go into the evaluation chapter.

3.3 [Background one]

If your work builds on prior work or research, this is the place where you can introduce the necessary knowledge to the reader. For instance, if you work on business process modeling and it is your goal to develop an extension of the modeling language BPMN, here you provide the necessary background knowledge so that the reader will be able to follow your subsequent discussions on the matter. Be cautious to introduce all and only those concepts, constructs, tools, languages that you really need.

3.4 [Background two]

If your work builds on more than one prior work or research, add respective sections. For instance, if your extension of BPM is meant to leverage on crowdsourcing to perform work, here you provide the necessary background on crowdsourcing.

Chapter 4

[Core contribution]: Approach

This is the chapter where you explain how you approach the problem and how you intend to meet the requirements identified in the previous chapter. In short, here you explain your *solution*. But attention: you won't be able to describe every aspect of your thesis project here, in one single chapter. You will need more than one for that. So, this is the chapter where you explain your solution in terms of the general approach and the design decisions that you make:

- ☐ Identify the target *actors* that will benefit from your solution, describe them.

4.1 Design Decisions

Discuss here your decisions and strategy. Defer the details to the following chapters, which you can use to elaborate better on the core aspects of your work. Decide which of the design decisions are easy to explain and do not need any further elaboration and which instead deserve an own chapter. For example, if you work on a modeling language and you introduce new modeling constructs, the modeling constructs represent one of the core contributions of your work, and they should be explained in a chapter on their own. Similarly, if you develop a new algorithm, the design of this algorithm may deserve an own chapter. The rule of thumb is that those aspects of your work that require most effort very likely deserve a chapter on their own with enough space to explain why they required such a lot of effort.

- ☐ Identify the core *design decisions* that must be taken, name them, and explain them to the reader.
- ☐ Discuss the different *options* that are available for each of these decisions, describe them and possibly discuss pros and cons.
- ☐ For each decision to be taken, *make your choice* and *motivate* your choices with suitable arguments.

4.2 Architecture

Describe here how your software prototype (if developed) is structured.

- ☐ Identify the core *artifacts* (models, software prototypes, languages, etc.) that are needed to go from the problem to your solution, name and describe them.
- ☐ Identify the most important *dependencies* among these artifacts and make them explicit.
- ☐ Put everything into context in some form of *functional architecture* of your solution (if your solution consists in a software prototype).

I explicitly call the architecture "functional architecture" to emphasize that you should not talk about technologies, code, frameworks, or similar here. What instead is needed here is an explanation of the:

- ☐ *software modules* that your prototype will leverage on (all software can be structured into small modules to split the internal logic into smaller, hopefully self-explaining elements),
- ☐ their *interconnection*,
- ☐ their *inputs and outputs* (the artifacts identified earlier),
- ☐ the *actors* involved in the execution of the software.
- ☐ Use one or more *figures* (illustrations) to clarify the above.

Figures and tables
You are an engineer, and using figures (illustrations) and tables to better convey your ideas should be an obvious practice you should have learned throughout your university career. If not, it's time now. Use illustrations, screen shots, sketches,

and so on to help the reader understand. Use tables to summarize complex text (for example, a profound analysis of the state of the art) or to format data in a readable fashion. Each time you use a figure or table, you must also (i) complement it with a so-called caption (a text right underneath or above it) to give it a title and a description and (ii) reference it from within the main text (never just place a figure somewhere without talking about it). If you use Latex, check your Latex documentation for how to use captions and references.

Chapter 5

[Solution aspect one]

Elaborate here better on the first aspect.

Chapter 6

[Solution aspect two]

Elaborate here better on the second aspect.

Chapter 7

Implementation and Evaluation

Yes, you got it: finally, let's talk technology! If you are an attentive reader, you will have noticed that so far I restrained from talking about technology and implementation stuff. And that was intentional: doing a thesis is first and foremost a *conceptual* effort, meaning an effort that should require a lot of brainwork, thinking, reasoning, discussing, drawing sketches of ideas, constructing tables for making informed choices, and so on.

And you know what? If that is well done and well described, your reader, even if he/she is not tech-savvy or an expert in your topic, will understand you and be able to follow your reasoning and agree/disagree with the choices you propose. If instead you start too early talking about technologies, programming languages, protocols, fancy frameworks that your reader does not know and, even worse, explain your solution in function of these technologies, you will loose the attention of your reader. And there is nothing as bad as that.

Once you loose the attention of your reader due to too much geek talk, you will not be able to get the attention back. The consequence is that, even if you did the best project ever and come up with Nobel Prize worthy findings, your reader (perhaps your reviewer) will not notice, and you will not get the credit you actually deserve.

The lesson learned is: *defer* the tech talk as long as possible (too early = too dangerous), *single it out* from the rest of the work (so that who is not interested in the low-level details can skip it), and make it *self-contained* (so that who instead wants to read it gets all the details necessary).

7.1 Implementation

Here you can describe the technologies you use, put code example, describe all the details you feel are needed to enable the reader (with the necessary tech background) to understand. The goal of this section should be to enable your reader to re-implement what you did, perhaps with different technologies.

- ☐ Describe the *technologies* you use in your solution.
- ☐ *Motivate* possible technology choices.
- ☐ Copy and paste here the *architecture figure* you should already have included in Section 4.2 and extend it with the technologies you use for each of the modules.
- ☐ Provide insight into the most important *implementation problems* and how you solve them.
- ☐ If available, provide a link to an *online repository* holding the code of your prototype (ideally released as open-souce software on GitHub or the like).
- ☐ Maybe you also want to share here some *UML diagrams* you drew before starting with the coding of the software.
- ☐ Provide evidence that your prototype *works*, e.g., screen shots, produced outputs, or similar.

7.2 Evaluation

This is a section that may be missing in a tesina, while for a thesis it is of fundamental importance. Even more: in some projects, the evaluation may even be a major contribution of the work and deserve an own chapter. If this is your case, then do so. For example, if you do an elaborated user study that requires careful literature study, design, planning, execution, data collection, data analysis, then you may want to make this effort also evident in the structure of the thesis by giving it an own chapter (remember that the structure of the thesis should already tell the reader a story).

7.2.1 Design of Evaluation

Explain here how you evaluate your solution, e.g., you do a controlled performance study in the lab using a cluster of 50 computers in a network, or

you do a simulation of an algorithm for which you first do some probing of some environment to fine-tune some parameters of the algorithm to have the simulation represent as real as possible situations, or you may do a user study, or... Here some options:

- ☐ *Theorem proofing*: if your work is of pure theoretical nature, you may want to accompany your theorems and corollaries with suitable proofs. Doing so requires good mathematical and/or algebraic skills.
- ☐ *Data analysis*: if you work on a topic that is related to Data Science, likely you will have a lot of data to analyze. Explain which data you are considering, how it is collected and prepared for the analysis, which kind of statistical analyzes you intend to use, why, etc.
- ☐ *Performance test*: if instead you develop a software prototype and claim that it works better than some exiting algorithm/software, explain which is your baseline to back your claim, tell how you want to compare your solution to the existing ones, which results you consider a success and which instead represent a failure, etc.
- ☐ *User study*: if your work involves real users in the evaluation of your work, explain how you select the participants, if they have to sign a consent form or not, if they need to obtain some form of prior training, which data you collect, how you guarantee their privacy and the security of the collected data, how you analyze the data, etc.
- ☐ *Simulation*: if you are not able to run your solution in a real environment and instead have to fall back to a simulation, explain how you set up the simulation environment, which assumptions you make, how you configure the simulation environment so that it resembles real situations, which exact data you collect, how you analyze it, etc.
- ☐ *Case study*: if the nature of your work does not allow a systematic data collection to back your claims, perhaps you want to elaborate on a case study that showcases the use of your solution in a real or fictitious application scenario. Explain the requirements of the case study, tell how realistic the case study is, show how your solution helps.
- ☐ ...

7.2.2 Metrics

Remember when I talked about the requirements and that ideally it would be good if the reader in the end of the thesis was able to use the list of

requirements as a checklist and to tick boxes? Well, this is where the reader should get the necessary tools to tick the boxes. Most likely, some of the requirements, claims and evaluation designs will need some specific metric to be able to tell if a requirement is satisfied or not. For example, you may want to measure response time for a time-critical service, or precision/recall for works on information retrieval, or individual quality attributes in crowdsourcing, or...

- ☐ Define all the *metrics* needed by your evaluation designs.
- ☐ Tell how to *assess* the requirements and claims of your thesis.

7.2.3 Results

In this subsection you report on the results of the experiments/evaluation you perform. Report on all the important numbers for each of the metrics, on possible issues with running the code, etc. This is however not yet the place where to go into lengthy considerations on the meaning of values, this is for the next subsection. It's good to explain comparative results (A better than B in condition X, while in condition Y B is better), outliers (in one very specific situation A has an extraordinarily low/high performance), general statistics.

7.2.4 Discussion

Finally, here you discuss your results. That is, you discuss the *meaning* and *impact* of your result for the goals of your thesis. In other words, you *interpret* the results in light of your goals, expectations, intuitions, hypotheses. Did the prototype meet the expected performance? Is the achieved statistical significance reached to draw conclusions you would not be afraid of defending in front of a commission? Was the problem solved? Too slow? Too fast? Give the reader a feeling (as well as convincing arguments and numbers) for why you think some requirements are met while others may be missed.

Chapter 8

Conclusion and Future Work

So far so good. We are almost done. What is left is, well, just one of the most important chapters of the whole thesis, i.e., the conclusion. The purpose of this section is not to "conclude" the thesis in the sense to "stop" here. It's rather to draw conclusions, that is, tell how well your work actually meets the requirements identified, answers the research questions, advances the state of the art. As such, this is perhaps the most important section! It may seem easy to just summarize a bit what you did and tell again what your objectives were when starting the work. But be aware that this can be much more difficult than it sounds, and you can expect your supervisor iterating with you several times over this same chapter. It is important that you show again your personal and professional maturity and your understanding of the topic. As you will see, some healthy self-criticism too is needed to make this chapter good.

8.1 Summary and Lessons Learned

Summarize here your work in about one page.

- ☐ Start from the initial *problem statement* or *research questions*.
- ☐ Summarize your *approach* and *methodology*.
- ☐ Recap the *lessons learned*.

8.2 Outputs and Contributions

Provide an overview of the outputs your project/work produced and then state what you think are the (research) contributions that advance the state of the art.

- ☐ List all the concrete *outputs* you produced (remember the discussion in Section 1.4).
- ☐ Copy/paste here the *list of contributions* you already anticipated in Section 1.4 (attention: outputs and contributions are two different lists; don't mix them).
- ☐ For each of the contributions, provide suitable *evidence*, drawing from the body of your thesis. For instance, if you claim that you did a formal proof of something, provide the exact number or name of the proof. If you promised subjective evidence for something, link this claim to the user studies you did. Etc. One or two sentences are enough for each of the contributions.

8.3 Limitations

This is where your self-criticism is needed. By now, I am confident you did a great work with your project and the writing of your thesis. So, compliments for that! You're almost done. But let's be frank: the work is not perfect. It simply cannot be, it never is. If it is, then not only I but also the whole commission of your defense will give you a standing ovation (I really would like to see this once). But in general there are just so many aspects of a research/thesis project that one would have to control or test, and with the limited time and resources available for these kinds of final projects it is just not possible to do everything.

In this section, you therefore tell the reader which aspects of your work may limit the impact or generalizability of your findings or contributions. As said, be frank. If you tell that you did a user study with only 10 people instead of 30 (which would make the findings stronger), you don't risk to give the impression you didn't do it well enough. Actually the opposite is true: if you don't tell it, your reader, who by now will anyway have gotten that there were 10 and not 30 people involved in the study, will instead think either (i) that you *didn't know* that a higher user involvement would have been better to back your claims or (ii) that you intentionally want to *hide* information or even *cheat*. None of these are good for you, and for sure worse then telling straightaway. Keep this in mind.

Here some typical limitations of research. Check if any of them apply to your work:

- ☐ Small *sample size* (e.g., the number of users in the study or the amount of data collected in an experiment).

- ☐ For experiments that involve multiple *indipendent variables*, likely you will not have tested them all (e.g., in a crowdsourcing experiment, you fixed a reward for all experiments and did not study if that too affected your results).
- ☐ You may have *promised* something in the beginning of the thesis; if you didn't achieve everything either you drop the very promise or you mention it here as a limitation.
- ☐ When you collected data, there may have been some *bias* in the data (e.g., if you implement a prototype and do a user study yourself where there participants know that you actually implemented the software, they will give you biased answers, typically better ones).
- ☐ Collected data many have turned out being *incomplete* or of *lower quality* then initially expected. How does this impact your findings?
- ☐ Your prototype may have *crashed* or *not worked properly* in some experiments; it's important you tell the reader and explain possible implications of this on the validity of your conclusions.
- ☐ Due to time restrictions, you may have *not been able to complete* all experiments planned initially; again, explain the possible implications.
- ☐ People participating in a user study may have *dropped out* of the study, for whatever reason; if the reason is related to what you did or not did, you should mention it.
- ☐ Sometimes it is *not possible to compare* an own algorithm with other, similar algorithms, e.g., because their code is not available; this too may limit the viability of the findings.
- ☐ ...

8.4 Future Work

Finally, here you tell the reader which aspects you think would deserve further study or development. A good starting point for this is of course the list of limitations you just discussed. Not all of them may be worth investing more effort, but some will. The idea of this section is to identify where possible new effort should be invested, in order to make the work complete. Again, be frank and don't be afraid of identifying also new research directions. It's not you who will be doing what you propose here. It's meant for

the reader, the community. Everybody understands that after your defense you won't be working any longer on this project. It's all about suggesting future work, not telling that *you* will be doing it.

Bibliography

[1] R Hevner Von Alan, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, 28(1):75–105, 2004.

Appendix A

User Manual

If you implemented a piece of software that is meant to be used by somebody else than you, then here you can provide a brief user manual that tells the target user how to use it. Part of this is the possible installation of the software and its operation and trouble shooting.

Appendix B

Dataset

If your work was based on a dataset that can be considered an output of the project, here you can describe it in detail.