Doing a Master's Thesis in Machine Learning or NLP

A Practical Guide

Finding a contribution, defending it, and what it becomes next

Konstantinos Skianis

Assistant Professor of Natural Language Processing
Department of Computer Science and Engineering
University of Ioannina

https://y3nk0.github.io

A master's thesis sits between an undergraduate project and a PhD chapter, and it is judged by a different standard than either. You are expected to make a *contribution* — something the reader did not know before — but a bounded one, finished inside two semesters. This guide is about choosing that contribution, running experiments that survive scrutiny, writing it up, and turning it into a paper, a PhD application, or a job.

October 2026

Your department's own regulations — length, format, deadlines, ECTS, grading — always take precedence over anything here.

Contents

1 What a master's thesis has to be **2**

2 Finding a contribution **2**
2.1 Five routes that work . . . 2
2.2 Where the open questions are written down . . . 3
2.3 The two-page proposal . . . 3

3 Industry, external and NDA-bound theses **4**

4 Related work, the chapter that separates degrees **4**

5 Tools, set up in week one **5**

6 Experiments that survive an examiner **5**
6.1 The order that works . . . 5
6.2 Claiming a difference honestly . . . 5
6.3 The mistakes that cost a grade . . . 6
6.4 Keep a results table from day one . . . 6

7 Using an AI assistant **6**

8 Writing it **7**

9 Turning the thesis into a paper **7**

10 The defence **8**

11 Working with your supervisor **8**

12 A timetable **8**

13 What it becomes next **9**

14 Checklists **9**

A Link directory **10**

B The two-page proposal, as a form **10**

1 What a master's thesis has to be

Three projects, three standards. It helps to see them side by side, because students routinely aim at the wrong one.

	Undergraduate	Master's
Question	Given, or chosen from a menu	**Yours**, argued for and scoped with your supervisor
Bar	Done correctly, understood completely	Correct *and* tells the field something it did not know
Related work	A summary of what exists	A **positioned** chapter: a taxonomy, and where your work sits in it
Evidence	Baselines, a fair comparison, error analysis	All of that, plus ablations, several seeds, and significance where it is claimed
Typical output	A working system and a report	A report that could become a **workshop paper** with a month's extra work
Length	30–50 pages	40–80 pages, by department rule

The one-claim test

Write the sentence your thesis exists to defend, in this shape:

> *For **[task/setting]**, **[what you did]** achieves **[effect]** compared with **[baseline]**, because **[mechanism]** — shown by **[the experiment]**.*

If you cannot fill every slot by the end of month two, you have a project, not a thesis. The missing slot tells you what to work on next: usually *because* (you have numbers but no explanation) or *compared with* (you never built the honest baseline).

What a PhD requires that you do not

Novelty at the level of a main-conference paper, a programme of work spanning years, and a survey of a whole subfield. Aiming there will cost you the deadline. A master's contribution can be: a method applied somewhere it has not been and evaluated properly; a careful reproduction that finds the limits of a published claim; an ablation that identifies which component actually matters; a new resource or benchmark for an under-served language or domain. Each of those is publishable at a workshop and finishable in a semester.

2 Finding a contribution

You need a question that is *open*, *answerable with your compute*, and *interesting whichever way the answer falls*. That last property is what protects you: if your idea works you have a result, and if it fails you have a finding, provided the experiment was fair.

2.1 Five routes that work

1. **Extend one paper.** The most reliable route. Take a recent method you admire and push exactly one dimension: a new language or domain, a harder distribution shift, a tenth of the compute, a tenth of the labels, longer inputs, a different evaluation. The baseline is the paper; the contribution is where it breaks and why.

2. **Reproduce and interrogate.** Rebuild a published result, then ask the question the authors did not: does it hold across seeds, datasets, model scales, prompts? Reproducibility work is publishable in its own right and teaches you more about experimental rigour than any course.

3. **A shared task, with an angle.** SemEval and the workshop tasks at *ACL venues hand you data, a metric and a leaderboard (§5 has the links). A master's thesis does not stop at the

ranking: the thesis is the analysis of *why* systems place where they do, which the system paper has no room for.

4. **An applied or industry thesis.** A real problem from a company, a lab in another department, or a public body. The research question has to survive contact with the deployment constraint — see §3 before you sign anything.

5. **Analysis or theory.** Probing what a representation encodes, a robustness study, an efficiency comparison measured honestly, a critique of a metric with a small human evaluation, or a bounded theoretical result. Keep the claim small and complete.

2.2 Where the open questions are written down

Six places to look, in order of yield

1. **Limitations sections.** Mandatory at ACL venues: authors list what their method cannot do. It is a menu of theses.
2. **Task overview papers.** The closing section says exactly what nobody solved this year.
3. **Survey taxonomies.** Find the most recent survey and look for the boxes with one or two entries.
4. **Workshop calls for papers.** This year's open problems, in bullet form, written by the people who will review you.
5. **Leaderboards that have plateaued**, or where a simple baseline sits suspiciously close to the top — a sign the task, metric or data deserves a second look.
6. **Your supervisor's current grant.** Unglamorous and effective: the open questions are already funded, the data already exists, and the supervision will be attentive because the work matters to them.

2.3 The two-page proposal

Before you write code, write two pages and get them signed off. This is longer than the undergraduate version because the risks are larger.

Section	What goes in it
Question	One sentence, open, answerable
Claim	The one-claim sentence above, with the slots filled as far as you can
Why now	Two or three references showing the gap is real and current
Data	Name, size, licence, access route, and whether you already have it
Baselines	The trivial one, the published one, and where its numbers come from
Method	What you will try, in three bullets, with the one knob you will vary
Evaluation	Metric, splits, how many seeds, and the test you will use to claim a difference
Compute budget	GPU hours you can actually get, and the largest model that fits
Milestones	Five dates: data, baseline, first result, freeze, full draft
Risks	The three most likely ways this stalls
Fallback	The thesis you still have if the main idea fails — usually the comparison and the analysis

Scope, the thing that actually sinks theses

Not difficulty — scope. Three questions, honestly: can you name the dataset and metric today? Can you run one baseline end to end within three weeks? If the main idea fails completely, is there still a thesis? Any "no" means narrow it now, while narrowing is free.

3 Industry, external and NDA-bound theses

A thesis done inside a company can be excellent — real data, real constraints, a mentor who cares. It can also become unexaminable. Settle these *in writing, before you start*, with your academic supervisor in the conversation.

The six questions to ask on day one

1. **Can the thesis be published and defended openly?** A thesis is an examined document; most universities require it to be readable by examiners and often archived publicly.
2. **Is there an embargo?** Many departments allow a delay of six or twelve months. Find out the rule and whether the company needs it.
3. **What may appear?** Method, yes; raw data, often no; absolute numbers, sometimes no. Relative results ("+3.1 F1 over the internal baseline") are usually negotiable when absolutes are not.
4. **Who owns the code and the IP?** Say explicitly what you may put on GitHub. Agree it before you write it.
5. **Who supervises what?** You need an academic supervisor responsible for the research standard and a company mentor for the problem. When they disagree, the degree requirements win.
6. **What is the fallback if access is withdrawn?** Reorganisations and policy changes happen mid-thesis. A public dataset you can fall back to turns a catastrophe into a chapter.

Writing around confidential data

If the data cannot be described, the thesis has to stand on the method and the protocol: describe the data by its *statistics and structure* (size, label distribution, domain, annotation process) rather than its content, and reproduce the core experiment on a public dataset so an examiner can check your reasoning against something they can see. Do this from the start — it is painful to retrofit in month eight.

Personal data brings its own duties: GDPR applies to scraped social media, user logs, recordings and surveys. Ask about ethics approval early; in some departments it is required and takes weeks.

4 Related work, the chapter that separates degrees

The undergraduate version summarises papers. The master's version *positions* your work. Organise by theme, not by paper, and end by saying what nobody has done — which is, conveniently, what you are about to do.

A bounded search you can finish in three weeks

1. **Seed set:** 5–10 anchor papers from your supervisor. Read them properly; their vocabulary becomes your queries.
2. **Search systematically** across Google Scholar, Semantic Scholar, the ACL Anthology, OpenReview and DBLP — and **log every query, date and hit count** in a spreadsheet. That log is what makes the chapter defensible rather than anecdotal.
3. **Snowball both ways:** backward through reference lists, forward through "cited by" and *search within citing articles*. Forward snowballing is how you find the last six months.
4. **Screen in two passes:** title and abstract, then full text. Expect to screen 200–400 and keep 60–100.
5. **Extract into a table**, one row per paper: key, venue, problem, method family, data, metric, headline result, compute, code, and whether it reproduces.
6. **Cluster the rows** until the groups stop changing. That is your taxonomy, and it becomes one figure and the chapter's skeleton.

Read in three passes (Keshav): five minutes to decide, an hour for figures and method, several hours only for the five papers your thesis depends on.

5 Tools, set up in week one

The same stack as every other guide in this series; set it up once and stop thinking about it.

- **Zotero** (https://www.zotero.org) with the browser connector and **Better BibTeX** (https://retorque.re/zotero-better-bibtex/): pinned citation keys and auto-export into your thesis `refs.bib`. The quickest way to fill it: enable Scholar's BibTeX links (*Settings* → *Bibliography manager*), then **Cite** → **BibTeX**. Check every entry — a bibliography full of "arXiv preprint" for published papers is the clearest sign nobody looked. Cleaner sources: the **ACL Anthology** (https://aclanthology.org), **DBLP** (https://dblp.org), or any DOI from the terminal with `curl -LH "Accept: application/x-bibtex" https://doi.org/<DOI>`.
- **LaTeX**, in **Overleaf** (https://www.overleaf.com) for supervisor comments, and `git` + `latexmk` locally. One sentence per line makes diffs readable; `latexdiff` shows your supervisor exactly what changed between drafts. *Prism shuts down on 1 January 2027 — do not start a thesis there, and export anything you have.*
- **One git repository** for code, configs, notes and the thesis. Commit daily.
- **Compute:** your lab's cluster or Slurm queue, department GPUs, or Colab and Kaggle notebooks for smaller runs. Ask in week one — accounts take time, and the largest model you can train is a design constraint, not a detail.
- **Experiment tracking** from the first run (Weights & Biases, MLflow, TensorBoard): config, metrics and the git commit, every time.
- **Shared tasks and deadlines:** https://semeval.github.io, https://paperswithcode.co/ai-deadlines and http://aideadlines.org/?sub=ML,CV,NLP.

6 Experiments that survive an examiner

6.1 The order that works

1. Get the data and read fifty examples by hand. Count labels, find duplicates, look for leakage between splits.
2. Build the **trivial baseline** — majority class, TF-IDF plus logistic regression, a rule — and measure it. It anchors everything.
3. Reproduce the **published baseline**. If you cannot match it, nothing downstream is interpretable, and finding out why is itself a result.
4. Only then your method, one change at a time, each measured.
5. **Ablate**: remove each component and show what it was worth. Start early; ablations always take longer than the headline run.
6. **Error analysis**: categorise 50–100 mistakes by hand. This is where the *because* in your one-claim sentence comes from.

6.2 Claiming a difference honestly

Seeds, spread and significance

- Run every configuration with at least **three seeds** — five if the budget allows — and report **mean** ± **standard deviation**. A single run that improves by 0.4 is noise, and an examiner will say so.
- When you claim a system is better, test it: a **paired bootstrap** or **permutation test** on the test predictions is standard in NLP, easy to implement, and makes no distributional assumptions.
- Comparing many systems on the same data inflates false positives — say so, or correct for it.
- **Report compute**: GPU type, hours, parameters, and the cost of the hyperparameter search. A method that wins only after a hundred trials against a baseline tuned for one is not winning.
- Fix the splits *before* you look at results, and touch the test set once.

Negative results, handled properly

"I tried X, it did not beat the baseline, here is the evidence and here is why" is a legitimate master's thesis — provided the comparison was fair, the implementation was checked against a reference, and the analysis explains the failure. Many strong theses are exactly this. What is not acceptable is a negative result from a bug you never looked for: before concluding, verify your implementation reproduces a known number on a known dataset.

6.3 The mistakes that cost a grade

Seen every year

- **Leakage.** Tuning on test, duplicates shared between splits, or preprocessing and feature selection fitted on the full dataset before splitting. It invalidates everything downstream.
- **An unfair comparison.** Your method tuned for a week, the baseline run on defaults for an hour. Examiners spot it immediately, and it is the fastest way to lose their trust in the rest.
- **A difference claimed from one seed.**
- **Accuracy on imbalanced data.** Report macro-F1, per-class results, or a confusion matrix.
- **The method chapter that cannot be reimplemented** because the details live only in the code.
- **A related-work chapter that is a list.** Paragraph per paper, no taxonomy, no positioning — the clearest marker of an undergraduate thesis wearing a master's cover.
- **Writing started in the last month.** It shows, and it is the single strongest predictor of a mediocre grade.
- **Results that cannot be regenerated** because the configuration lived in your shell history.

6.4 Keep a results table from day one

One CSV or spreadsheet: date, commit, config, seed, dev score, test score (only when you finally run it), and a one-line note on what changed and why. Every table and figure in the thesis should regenerate from logged results with one script. In month eight, "which run produced Table 4?" must have an answer.

7 Using an AI assistant

An assistant removes mechanical work and can quietly remove your degree. Both are true.

- **Good uses:** explaining unfamiliar notation, getting someone else's repository to run, boilerplate and plotting code, refactoring, LaTeX debugging, tightening prose you wrote, and acting as a hostile examiner on your draft.
- **Set it up** in the thesis repository with a short memory file of your conventions, connected to your files and your Zotero library, and keep a dated `notes/` log it can read.

Five rules

1. **You must be able to explain every line you submit.** The defence tests precisely this.
2. **Never cite a paper you have not opened.** Models invent plausible references; fabricated citations end theses.
3. **Never let it write your results or your analysis.** Numbers come from runs you did; interpretation is the degree.
4. **Follow your university's policy** and declare assistance where a declaration is required. Ask your supervisor; do not guess.
5. **Nothing confidential into a cloud tool** — company data under NDA, personal data, or someone else's unpublished manuscript.

8 Writing it

Start in month one. Writing is how you discover the hole in the argument, and holes are cheap to fix early and ruinous to find in the last fortnight.

Chapter	What goes in it	Pages
Abstract	Problem, what you did, headline result, in one paragraph	1
Introduction	Context, the question, your contributions as a numbered list, thesis outline	4–6
Background	Only the concepts your chapters actually use	6–12
Related work	By theme, with a taxonomy figure; ends by positioning your work	8–12
Method	Precise enough to reimplement from the text alone	8–12
Experiments	Data, splits, metrics, baselines, protocol, then results	10–18
Analysis	Ablations, error analysis, significance, what surprised you	6–10
Conclusion	Findings, limitations, future work	2–4
References, appendix	Bibliography; hyperparameters, prompts, extra tables	—

Sizes are a sane default for a 40–80 page thesis; your department's rules win.

- **Write the figures and tables first**, then the text explaining them. A table that needs three paragraphs of excuses means the experiment is wrong, not the prose.
- **Every chapter** opens with a sentence saying what it does and closes with one saying what it showed.
- **Contributions as a numbered list** in the introduction, each one checkable against a chapter.
- **Captions are self-contained**: a reader who looks only at your figures should still follow the thesis.
- **Cite properly, quote almost never.** Copying sentences — from papers, from the web, or from a model — without quotation marks and a citation is plagiarism, and it is checked.
- **Limitations, written honestly**, are a sign of maturity. Examiners find them anyway; better that you found them first.
- **A complete draft to your supervisor three weeks before the deadline.** Feedback on a partial draft is feedback on the wrong problem.

9 Turning the thesis into a paper

This is the step that makes a master's thesis count twice — as a degree and as a publication — and most students skip it because they are exhausted. Talk to your supervisor about it in month five, not after the defence.

- **What to cut:** the background chapter, most of the related work, the narrative of what you tried. A paper keeps the claim, the method, the key experiments and the analysis — eight pages, not sixty.
- **Where to send it:** a workshop at an *ACL or ML venue is the realistic first target; **Findings** and the **Student Research Workshop** are real, citable venues; a shared-task **system paper** if you took that route. The PhD guide in this series covers the venue landscape and the ACL Rolling Review.
- **Authorship:** agree the order early, in conversation, not by assumption. Typically you are first author and your supervisor last.
- **Timing:** the thesis deadline and the venue deadline rarely align. Submit the paper after you submit the thesis, using the examiners' comments as free reviews.

- **Release the artefacts:** code with a licence and a README that reproduces your main table, checkpoints on Hugging Face, and a `CITATION.cff` so people can cite it.

10 The defence

Usually 20–30 minutes plus questions, before a committee that has read the thesis unevenly.

- **Structure:** problem (2 slides) → why it is hard and what exists (2–3) → your claim (1) → method (3–5) → results (4–5) → analysis and limitations (2–3) → conclusion (1). One idea per slide.
- **Rehearse out loud, timed, twice**, at least once to a person.
- **Know your numbers:** dataset sizes, your result, the baseline, the spread across seeds, training cost.
- **Expect:** why this baseline? is the improvement significant? what happens under distribution shift? why does it fail on class X? what would you do with six more months? what did you find hardest?
- **"I do not know"** is a fine answer when followed by "here is how I would find out". Guessing is not.

11 Working with your supervisor

At master's level the division of labour shifts: your supervisor sets the direction and the standard, and you own the execution. The students who get the most from supervision are not the ones who need the least help — they are the ones who ask precise questions.

> **Decide yourself, bring to the meeting, escalate immediately**
>
> **Decide yourself:** library and framework choices, code structure, plotting, which of two equivalent baselines to run first, how to organise your notes. Bringing these to a meeting wastes the half hour you have.
>
> **Bring to the meeting:** anything that changes the claim or the scope — a result that contradicts the plan, a baseline you cannot reproduce, a dataset that turns out to be unusable, an idea that would add a month.
>
> **Escalate immediately, do not wait for the meeting:** losing data or compute access, an ethics or NDA problem, a deadline you are going to miss, and anything about your health or circumstances that affects the timeline. Supervisors can nearly always help with these *if told early* and almost never at the end.

- **Send the agenda the evening before** — five bullets, as below. A supervisor who has seen your numbers before the meeting gives you an hour of thinking instead of ten minutes of catching up.
- **Disagree in writing, with evidence.** "I ran the experiment you suggested; here is why I think the other route is better" is a good conversation. Silently doing something else is not.
- **Ask for the standard explicitly**: ask to see a thesis they considered excellent, and what separated it from an average one.
- **Say when you are stuck.** Two days stuck is normal; two weeks stuck and silent is a supervision failure that is half yours.
- **What a supervisor cannot do:** write it for you, guarantee a result, or rescue a thesis in the final fortnight. Plan accordingly.

12 A timetable

A nine-month plan; compress to six by halving months 2–4, and put every milestone in a calendar.

Month	Done by the end of it
1	Topic agreed; two-page proposal signed off; tools installed; data in hand and inspected; 15–20 papers read; trivial baseline running.
2	Published baseline reproduced; experiment tracking in place; search log running; related-work taxonomy drafted.
3–4	Method implemented; first real results; method chapter drafted; one-claim sentence filled in completely.
5	Ablations and seeds; significance tests; publication plan agreed with your supervisor; experiments chapter drafted.
6–7	Error analysis; analysis chapter; results frozen; all tables regenerated by script.
8	**Complete draft** to your supervisor; artefacts cleaned and released.
9	Revisions; every citation checked; defence slides; two rehearsals; submission.

> **Weekly rhythm**
>
> Meet weekly, and send five bullets the evening before: **what I did** · **what I found** · **what is blocked** · **what I propose next** · **what I need from you**. Keep a shared running document. "I am stuck" sent on Tuesday is worth three weeks of silence.

13 What it becomes next

A PhD application. Your thesis is the main evidence that you can do research. Give your referees the abstract, the results and your plans; ask while the work is fresh. A workshop paper from the thesis moves you from "promising" to "has published".

A job. The repository is the portfolio: a README that reproduces your main table, honest limitations, and clean code say more than any bullet point on a CV. Expect to be asked, in interviews, exactly what your examiners asked.

Both. They are not exclusive, and an industry year before a PhD is a perfectly good path.

14 Checklists

Month 1

- Supervisor meeting slot fixed; agenda format agreed
- Two-page proposal signed off
- Zotero + Better BibTeX; auto-export into `refs.bib`
- Git repository with `notes/` and a README
- Data obtained; licence and ethics checked
- Compute access working; a real job submitted
- Trivial baseline measured
- If industry: the six questions answered *in writing*

Before freezing results

- Trivial baseline, published baseline and your method on identical splits
- Three or more seeds; mean ± std reported everywhere
- Significance test for every comparison you claim
- Compute and hyperparameter search reported for all systems
- Test set used once; no tuning on it
- Ablations complete; errors categorised by hand
- Every table and figure regenerated by script from logged results
- Code runs from a clean clone, following your own README

Before submitting

- One-claim sentence appears, in plain words, in the abstract and the conclusion
- Contributions list in the introduction, each checkable against a chapter
- Every citation checked against the real paper — no invented references
- Limitations written honestly
- Department format, length and declaration requirements met
- Complete draft seen by your supervisor at least three weeks earlier
- Slides rehearsed twice, timed; demo recorded as a fallback
- Publication plan agreed: venue, authorship, timing

A Link directory

What	Where
Literature	
Google Scholar	https://scholar.google.com
Semantic Scholar	https://www.semanticscholar.org
ACL Anthology	https://aclanthology.org
DBLP	https://dblp.org
OpenReview	https://openreview.net
Connected Papers	https://www.connectedpapers.com
Papers with Code	https://paperswithcode.co
Tasks and deadlines	
SemEval	https://semeval.github.io
Conference deadlines	https://paperswithcode.co/ai-deadlines
	http://aideadlines.org/?sub=ML,CV,NLP
Tools	
Zotero	https://www.zotero.org
Better BibTeX	https://retorque.re/zotero-better-bibtex/
Overleaf	https://www.overleaf.com
Hugging Face	https://huggingface.co
Weights & Biases	https://wandb.ai
The companion guides	
Undergraduate thesis, PhD	https://y3nk0.github.io/guides/

B The two-page proposal, as a form

Title. Working title.

Question. The one sentence this thesis answers.

Claim. *For* _____, _____ *achieves* _____ *compared with* _____, *because* _____ — *shown by* _____.

Why now. Two or three references showing the gap.

Route. Extend a paper / reproduce / shared task / applied / analysis (§2).

Data. Name, size, licence, access, already in hand?

Baselines. Trivial; published; source of its numbers.

Method. Three bullets, and the one knob you vary.

Evaluation. Metric, splits, seeds, significance test.

Compute. Hardware, hours, largest model that fits.

Milestones. Data _____; baseline _____; first result _____; freeze _____; draft _____.

Risks. Three ways this stalls.

Fallback. The thesis you still have if the idea fails.

If industry. Publication, embargo, what may appear, IP, supervision, fallback access.

The shortest version of all of this

Write the one-claim sentence in month one and let it tell you what is missing. Reproduce the baseline before you innovate. Change one thing at a time, with enough seeds to know it is real. Write from the first month, release what you built, and plan the paper before you are exhausted. A bounded contribution, finished and defended, beats an ambitious one abandoned in month eight.