# Doing a Master's Thesis in Machine Learning or NLP

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

y3nk0.github.io

October 2026 ▪ companion to the full guide and the one-page cheat sheet

# Three projects, three standards

| | Undergraduate | Master's |
|---|---|---|
| Question | Given, or from a menu | **Yours**, argued for and scoped |
| Bar | Done correctly, understood | Correct *and* tells the field something it did not know |
| Related work | A summary of what exists | A **positioned** chapter: a taxonomy, and where you sit in it |
| Evidence | Baselines, fair comparison, error analysis | All that, plus ablations, seeds, and significance where claimed |
| Output | A system and a report | A report that could become a **workshop paper** in a month |
| Length | 30–50 pages | 40–80 pages |

A PhD asks for a *programme* of work. You are asked for *one bounded contribution*, finished.

# The one-claim test

## Write the sentence your thesis exists to defend

*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 empty slot is your next task

Usually *because* — you have numbers but no explanation — or *compared with* — you never built the honest baseline.

Part 1

# Finding a contribution

# Five routes that work

- **1. Extend one paper.** The most reliable. Take a recent method and push *one* dimension: a new language or domain, a harder distribution shift, a tenth of the compute or labels, longer inputs, a different evaluation. The contribution is where it breaks and why.
- **2. Reproduce and interrogate.** Rebuild a published result, then ask what the authors did not: other seeds, datasets, scales, prompts.
- **3. A shared task, with an angle.** SemEval and the workshop tasks hand you data, a metric and a leaderboard. The thesis is the analysis of *why* systems place where they do.
- **4. An applied or industry thesis.** A real problem, with the research question surviving contact with the deployment constraint.
- **5. Analysis or theory.** Probing, robustness, efficiency measured honestly, a metric critique with a small human evaluation, or a bounded proof.

Pick a question that is **open**, **answerable with your compute**, and **interesting whichever way it falls**.

# Where the open questions are written down

1. **Limitations sections** — mandatory at ACL venues. A menu of theses, in the authors' own words.
2. **Task overview papers** — the closing section says what nobody solved this year.
3. **Survey taxonomies** — the boxes with one or two entries.
4. **Workshop calls for papers** — this year's open problems, written by the people who will review you.
5. **Plateaued leaderboards**, or a simple baseline sitting suspiciously close to the top.
6. **Your supervisor's grant** — unglamorous and effective.

## Why the last one works

The questions are already funded, the data already exists, and the supervision will be attentive, because the work matters to them.

# The two-page proposal, before any code

| | |
|---|---|
| Question | One sentence, open, answerable |
| Claim | The one-claim sentence, slots filled |
| Why now | 2–3 references showing the gap |
| Data | Name, size, licence, already in hand? |
| Baselines | Trivial; published; source of its numbers |
| Method | Three bullets, and the knob you vary |
| Evaluation | Metric, splits, seeds, significance test |
| Compute | Hours you can get; largest model that fits |
| Milestones | Data, baseline, first result, freeze, draft |
| Risks, fallback | The thesis you still have if it fails |

## Scope sinks theses, not difficulty

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.

Part 2

# Industry theses, and related work

# If the thesis happens inside a company

## Settle in writing, before you start

1. Can the thesis be **published and defended openly**?
2. Is there an **embargo**? Six or twelve months is common.
3. **What may appear?** Method yes; raw data often no; relative results usually negotiable when absolutes are not.
4. Who owns the **code and IP**, and what may go on GitHub?
5. **Who supervises what?** When academic and company advice conflict, degree requirements win.
6. What is the **fallback** if access is withdrawn?

### Writing around confidential data

Describe the data by its *statistics and structure* — size, label distribution, domain, annotation — not its content, and reproduce the core experiment on a public dataset so an examiner can check your reasoning.

Do it from the start; retrofitting in month eight is painful.

GDPR and ethics approval: ask in week one. It takes weeks.

# Related work: the chapter that separates degrees

Undergraduate summarises. Master's **positions**: by theme, not by paper, with a taxonomy figure, ending where nobody has been.

1. 5–10 anchor papers from your supervisor; steal their vocabulary.
2. The same queries across Scholar, Semantic Scholar, the Anthology, OpenReview, DBLP — **log query, date, hits**.
3. Snowball backward (references) and forward (cited-by, *search within citing articles*).
4. Screen 200–400, keep 60–100; extract into a table.
5. Cluster the rows until the groups stop changing — that is your taxonomy.

## Read in three passes

5 minutes to decide → 1 hour for figures and method → hours only for the five papers your thesis depends on.

## The log is the point

A recorded search is what makes the chapter defensible rather than anecdotal — and it is the first thing a careful examiner probes.

Part 3

# Evidence

# The order that works

1. Get the data; read **fifty examples by hand**; look for leakage between splits.
2. Build the **trivial baseline** and measure it. It anchors everything.
3. **Reproduce the published baseline.** If you cannot, finding out why is itself a result.
4. Only then your method — one change at a time, each measured.
5. **Ablate.** Start early; they always take longer than the headline run.
6. **Error analysis**, 50–100 cases by hand. This is where the *because* comes from.

## A results table from day one

Date, commit, config, seed, dev score, and one line on what changed and why. Every table in the thesis regenerates from it with one script.

In month eight, "which run produced Table 4?" must have an answer.

## Negative results count

If the comparison was fair *and* you verified your implementation reproduces a known number first.

# Claiming a difference honestly

- **Three seeds minimum**, five if the budget allows; report mean $\pm$ standard deviation. A single run that improves by 0.4 is noise.
- **Test it:** a paired bootstrap or permutation test on the test predictions — standard in NLP, easy to implement, no distributional assumptions.
- Comparing many systems on one dataset **inflates false positives**. Say so, or correct for it.
- **Report compute:** GPU type, hours, parameters, and the size of the hyperparameter search.
- Splits fixed *before* you look; test set touched once.

## The comparison that is not a win

A method that wins only after a hundred search trials, against a baseline tuned once, is not winning. Examiners check this, and it is the fastest way to lose their trust in everything else.

# The mistakes that cost a grade

Seen every year

- **Leakage** — tuning on test, duplicates across splits, or preprocessing fitted before the split.
- **An unfair comparison** — your method tuned for a week, the baseline on defaults for an hour.
- **A difference claimed from one seed.**
- **Accuracy on imbalanced data** — report macro-F1 or per class.
- **A method chapter nobody could reimplement**, because the details live only in the code.
- **A related-work chapter that is a list** — the clearest marker of an undergraduate thesis wearing a master's cover.
- **Writing started in the last month.** It shows, and it predicts the grade.

# Tools, and the AI rules

## Week one

**Zotero** + Better BibTeX, auto-exporting into `refs.bib`; check every Scholar entry, and prefer the Anthology, DBLP or a DOI.

**Overleaf** plus `git` + `latexmk` locally, one sentence per line. *Prism closes 1 January 2027 — do not start there.*

One git repository; compute sorted in week one; every run tracked with its config and commit.

## Five rules for the assistant

1. **Explain every line you submit** — the defence tests exactly this.
2. **Never cite a paper you have not opened.** Fabricated references end theses.
3. **Never let it write your results or your analysis.**
4. **Follow your university's policy** and declare use where required.
5. **Nothing confidential** into a cloud tool.

Part 4

# Writing, defending, and after

# The shape of the thesis

| Chapter | What goes in it | Pages |
|---|---|---|
| Abstract | Problem, what you did, headline result | 1 |
| Introduction | The question; contributions as a numbered list | 4–6 |
| Background | Only the concepts your chapters use | 6–12 |
| Related work | By theme, with a taxonomy; ends by positioning you | 8–12 |
| Method | Precise enough to reimplement from the text | 8–12 |
| Experiments | Data, splits, metrics, baselines, protocol, results | 10–18 |
| Analysis | Ablations, errors, significance, surprises | 6–10 |
| Conclusion | Findings, limitations, future work | 2–4 |

Figures and tables first, then the text that explains them. Start in month one. **A complete draft three weeks before the deadline.**

# Getting what you need from supervision

**Decide yourself** libraries, code structure, plotting, which equivalent baseline to run first.

**Bring to the meeting** anything that changes the claim or the scope: a contradictory result, a baseline you cannot reproduce, an idea that would add a month.

**Escalate immediately** lost data or compute, an ethics or NDA problem, a deadline you will miss, anything about your circumstances. These are nearly always fixable *early* and almost never at the end.

## Five bullets, the evening before

what I did · what I found · what is blocked · what I propose · what I need from you

Ask to see a thesis they thought was excellent, and what separated it from an average one.

Two days stuck is normal. Two weeks stuck and silent is a supervision failure that is half yours.

# Turning it into a paper

- **What to cut:** the background chapter, most of the related work, the narrative of what you tried. Keep the claim, the method, the key experiments, the analysis — eight pages, not sixty.
- **Where:** a workshop at an *ACL or ML venue, **Findings**, the **Student Research Workshop**, or a shared-task system paper.
- **Authorship:** agree the order early, in conversation.
- **Timing:** submit after the thesis, using the examiners' comments as free reviews.
- **Release:** code with a licence and a README that reproduces your main table, checkpoints, and a `CITATION.cff`.

## Plan it in month five

Not after the defence. Most students skip this step because they are exhausted, and it is the step that makes the thesis count twice — as a degree and as a publication.

# The defence

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

- Problem (2) → why it is hard and what exists (2–3) → **your claim** (1) → method (3–5) → results (4–5) → analysis and limitations (2–3) → conclusion.
- Rehearse out loud, timed, twice; at least once to a person.
- Know your numbers: data sizes, your result, the baseline, the spread across seeds, the 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 "but here is how I would find out". Guessing is not.

# Nine months, and what comes after

| | |
|---|---|
| **1** | Proposal signed off; data in hand; 15–20 papers; trivial baseline |
| **2** | Published baseline reproduced; tracking; taxonomy drafted |
| **3–4** | Method; first results; claim sentence complete |
| **5** | Ablations, seeds, significance; publication plan agreed |
| **6–7** | Error analysis; results frozen; tables by script |
| **8** | **Complete draft**; artefacts released |
| **9** | Revisions; citations checked; two rehearsals; submit |

## What it becomes

**A PhD application** — the thesis is your evidence; a workshop paper moves you from "promising" to "has published".

**A job** — the repository is the portfolio, and interviews ask what examiners ask.

**Both** — an industry year before a PhD is a good path.

## The shortest version

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.

Konstantinos Skianis ▪ University of Ioannina ▪ y3nk0.github.io