Undergraduate Thesis in ML / NLP

The one-page version — topic, tools, experiments, writing, defence

Konstantinos Skianis
Assistant Professor of NLP · Dept. of Computer Science & Engineering
University of Ioannina · y3nk0.github.io

WHAT A THESIS IS

A supervised project, 4–9 months, that shows you can take a research-level problem, work methodically, and explain the result.

Examiners look for a clear question · correct methodology (sane baselines, honest evaluation, no leakage) · **understanding** — you must explain every design decision and every line you submit · clear writing · honest limitations.

Not required beating the state of the art, a new architecture, or training a big model from scratch. A careful reproduction with analysis is a good thesis; a record chase ending in one unexplained number is not.

> **Scope test — the usual failure.** Can you name the **dataset** and **metric** today? Can you run **one baseline** within **two weeks**? If your main idea fails completely, is there still a thesis? Any "no" = narrow it with your supervisor before writing code.

FINDING A TOPIC: THREE ROUTES

You do not need an original idea — you need a well-defined problem with data and a way to measure success. **Check advertised topics first:** most supervisors list open ones on their page or the department board. Pick one and do a small piece of research before the first meeting (key papers, dataset, metric, three questions). Only if you are free to choose, use the routes below.

A. Shared task / challenge Enter an organised competition: data, metric, baseline, leaderboard and deadline already exist. **The strongest default.**

B. Application thesis Take a method that works to a new language (Greek), domain (legal, medical, financial, code), or constraint (runs on a laptop, <100 ms, 50 labels, no cloud API).

C. Analysis / theory Reproduction and stress-testing · ablation: which of the five components actually matters · robustness (typos, dialect, domain shift, long inputs) · probing and interpretability · efficiency (distillation, quantisation, pruning) · critique of the metric with a small human evaluation · a small, complete theoretical result.

WHY A SHARED TASK FITS SO WELL

- The hardest parts — problem definition, data, annotation, evaluation protocol — are settled.
- A hidden test set means you cannot fool yourself.
- Organisers publish a baseline; past editions give a whole comparison table free.
- A deadline: the best project-management tool ever invented.
- A **system description paper** is peer-reviewed, archival and achievable — often your first paper.
- "4th of 31 teams" is concrete on a CV.

Where to find them SemEval (dozens of tasks a year) · workshop shared tasks at *ACL venues — WMT, CoNLL, BabyLM, BioNLP, WNUT, CheckThat! (see ACL Anthology + each conference's workshop list) · CLEF labs and TREC tracks (BioASQ) · NeurIPS competition track · Kaggle, CodaBench, EvalAI, AIcrowd, DrivenData · Hugging Face leaderboards.

> **Check the dates first.** Registration → data release → dev phase → a short evaluation window → system paper. Dates change every edition: read the task page and confirm they fit your thesis deadline. **Window closed?** Run it retrospectively — past data, metrics and the published participant table are usually public.

Running it as a thesis Read last year's *task overview* + 3–4 system papers (30 min each) · reproduce the organisers' baseline *first* · build your own validation split and trust it · log every run · submit something weak early — pipelines always break · the thesis is the long version of the system paper.

WHERE GAPS ARE WRITTEN DOWN

1. **Limitations sections** — mandatory at ACL venues: authors list what their method cannot do. A topic menu.
2. **Future work** in recent papers and surveys.
3. **Survey taxonomies** — find the boxes with one or two entries.
4. **Workshop calls for papers** — this year's open problems, in bullet form.
5. **Task overview papers** — the closing section says what nobody solved.
6. **Plateaued leaderboards**, or a simple baseline suspiciously close to the top.

Also: GitHub issues ("cannot reproduce Table 2", "does not work for language X") and failed reproducibility reports.

THE ONE-PAGE PROPOSAL

Write it before any code; send it to your supervisor. **Question** (one sentence, with a verb) · **why it matters** (two sentences, no hype) · **data** (name, size, licence) · **metric** (official + what else you report) · **baselines** (trivial + strong) · **method** (three bullets) · **compute** (and whether it suffices) · **milestones** (baseline, first result, freeze, draft) · **risks + fallback** (the thesis you still have if it fails).

TOOLKIT, WEEK ONE

References Zotero + connector + Better BibTeX: pinned keys (`devlin2019bert`), auto-export into your thesis `refs.bib`. One-line note on every save · Zotero flags retracted items · Unpaywall finds free copies of paywalled PDFs.

Fastest `.bib` entry Scholar *Settings* → *Bibliography manager* → show BibTeX links; then **Cite** → **BibTeX** → paste into `refs.bib` (or Zotero *Import from clipboard*); bulk-export from *My library*. **Check it** — "arXiv preprint" for a published paper is the usual error. Cleaner: Anthology / DBLP per-paper BibTeX, or any DOI from the terminal: `curl -LH "Accept: application/x-bibtex" https://doi.org/<DOI>`. Always cite the **published** version · protect capitals (`{BERT}`) · run `bibtex-tidy` and check every DOI resolves before submitting.

Writing Overleaf (co-editing + comments with your supervisor, a metered AI assistant; check for a university licence) · Prism **closes 1 Jan 2027** — do not start there; export now · locally `git` + `latexmk -pdf -pvc`, **one sentence per line**, `\citet` vs `\citep` · start from your department's template.

Code & compute one git repo for everything, commit daily · free GPUs: Colab, Kaggle Notebooks (longer sessions) · ask about lab machines in week one — accounts take time · track runs (W&B/MLflow) with config, metrics, commit hash · check model/dataset **licences** · personal data ⇒ talk to your supervisor before collecting (GDPR is not optional).

SEARCH & JUDGE THE LITERATURE

`"exact phrase"`	quote technical terms
`a OR b`	must be capitalised
`-survey`	exclude a term
`source:"SemEval"`	search inside a venue or task
`intitle:x`	precision filter
Cited by → search within citing articles	who applied it to *your* setting

Set 2–3 alerts; triage into Zotero weekly. Also Scholar Labs (ask the full question; it explains each hit) · ACL Anthology · Semantic Scholar · Connected Papers (week one) · OpenReview (read the reviews) · paperswithcode.co.

Read in 3 passes 5 min skim and decide → 1 h (figures, method) → hours (re-derive). Pass 3 for the five papers your thesis depends on.

Is this repo worth it? last commit, contributors, who answers issues · **search issues for "reproduce"** before trusting its numbers · checkpoints, pinned deps, **LICENSE** · stars measure marketing, not correctness · citations per year > raw totals.

AI ASSISTANT: HONEST USE

Assistant in the repo + a short memory file of your conventions · connect it to your files and Zotero · dated `notes/`. Use for: unfamiliar notation, getting someone's repo to run, boilerplate and plots, refactoring, LaTeX errors, tightening prose you wrote, a hostile-examiner pass · a free first review of your PDF from Deep Review (two a week; your supervisor still reads it).

> **1 You must explain every line you submit** — the defence tests exactly this. **2** Never cite a paper you have not opened: fabricated references fail theses. **3** Never let it write your results or your analysis. **4** Follow your university's policy and declare use if required — ask, do not guess. **5** No confidential or personal data in a cloud tool.

DOING THE WORK

1. Get the data; read **50 examples by hand**; count labels, find duplicates.
2. Dumbest possible baseline (majority class, TF-IDF + logistic regression, a rule) — this anchors everything.
3. Reproduce the strong baseline.
4. Only then your idea — **one change at a time**, each measured.
5. Ablate: remove each piece, show what it was worth. Start early.
6. Error analysis: categorise 50–100 mistakes by hand.

> **Mistakes that cost marks:** test-set leakage (tuning on test, duplicates, preprocessing fitted before the split) · one seed (+0.4 is noise — report mean ± std over 3+) · unfair comparison (your method tuned a week, baseline on defaults) · accuracy on imbalanced data (use macro-F1 / per class) · results you cannot regenerate · plots without axis labels or a caption that states the conclusion.

Results table from day one date, commit, config, seed, dev score, note on what changed. The results chapter is then formatting, not archaeology. **Negative results are results** — "I tried X, it did not help, here is the evidence and why" is a legitimate finding.

WRITING THE THESIS

Start in month one. Writing is how you find the hole in the story.

Abstract	problem, what you did, resul	1
Introduction	question + contributions list	3–5
Background	only what the reader needs	5–10
Related work	by *theme*, not one para per paper	5–8
Method	precise enough to reimplemen	5–10
Experiments	data, splits, metrics, baselines	8–15
Analysis	ablations, errors, surprises	4–8
Conclusion	findings, limitations, future work	2–3

Pages are a sane default — your department's rules win.

Advice figures and tables first, then the text · each chapter opens with what it does and closes with what it showed · past tense, active voice · captions self-contained · cite properly, quote almost never — copying without quotes and a citation is plagiarism, and it is checked · give your supervisor a **complete** draft 3 weeks before the deadline.

THE DEFENCE

15–25 min + questions. Problem (2 slides) → why hard (1) → what you did (3–5) → results (3–4) → analysis and limits (2) → conclusion. Rehearse out loud, timed, twice · a 60-second demo, recorded as a fallback · know your numbers (data sizes, your score, the baseline, training time). **Expect:** why this baseline? what if the data changes? why does it fail on class X? what with six more months? is the improvement significant? "I do not know, but here is how I would find out" is a fine answer.

SIX-MONTH TIMETABLE

M1 topic + proposal signed off, tools installed, data inspected, 10–15 papers, trivial baseline. **M2** strong baseline reproduced, logging, background + related work drafted. **M3** your method, first results, method chapter. **M4** ablations and seeds, tables generated by script. **M5** error analysis, **complete draft**, results frozen. **M6** revisions, references checked one by one, slides, two rehearsals, submit.

> **Weekly, to your supervisor:** what I did · what I found · what is blocked · what I propose · what I need from you. "I am stuck" sent on Tuesday is worth three weeks of silence.

AFTER THE THESIS

Publish it if it is good (system paper, workshop, student research workshop) · clean the repo: README that reproduces your main table, env file, licence — the best entry-level CV item · put up `username.github.io` · ask for the reference letter while it is fresh · thinking about a PhD? see the companion guide.

Companion to the full guide *Doing an Undergraduate Thesis in Machine Learning or NLP* · October 2026.