Assignment A3 : Discriminative Fine-tuning

Individual assignment • 20% individual-report component Released: 3 Dec 2026 Due: 10 Dec 2026, 23:59

In class we went from bag-of-words to embeddings to attention. In this assignment you will build the two ends of that ladder for **toxicity detection** — a classic **bag-of-words** classifier and a **fine-tuned transformer** — and measure the gap. The question to answer: *does the transformer earn its extra cost, and where does it help that bag-of-words cannot?*

Data. We use the comment text from the Kaggle competition *Jigsaw Unintended Bias in Toxicity Classification* (https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification). Load it with `load_textclf()` from the `starter/` bundle (join the competition first and set up Kaggle credentials). *Content note: the comments can be offensive.* You get `train` / `validation` / `test` splits (binary toxicity label + identity subgroups). Keep the configs small (`max_len`=128, 2–3 epochs): DistilBERT then fine-tunes in under ∼30 min on a laptop (Apple-Silicon `mps` or CPU); Colab is an optional fallback.

1. Tasks

1. **Baseline.** Train a bag-of-words classifier — `TfidfVectorizer` + `LogisticRegression` — and predict the `test` split.
2. **Fine-tune a transformer.** Fine-tune `distilbert-base-uncased` on `train` (tune on `validation`) and predict the `test` split.
3. **Compare.** Report macro-F1 of both and explain, with examples, where fine-tuning helps that bag-of-words misses. **[optional]** If you want, add a middle rung: pooled sentence embeddings + logistic regression.

What to submit & how it's graded. Submit `a3_pred_bow.csv` and `a3_pred_bert.csv` (columns `id,pred`; add `a3_pred_emb.csv` for the optional rung). Each is scored by macro-F1 against the hidden `test` labels; to pass, **DistilBERT must beat the bag-of-words baseline** by the posted margin. *Metric: macro-F1.*

2. Reflection (required, max 200 words)

Submit `reflection.md` (or `.pdf`). It must answer these three fixed questions and include the AI-use declaration. The reflection is graded for honesty and insight, not length.

1. **What did you know before?** — your starting point on this topic.
2. **What was hardest, and how did you get unstuck?**
3. **What does your result say, and what would you try next?**

AI-use declaration. State which parts (if any) were generated with AI tools, and *how you verified them.* Using AI is allowed; undisclosed or unverified use is not.

3. Submission

Upload a single `zip` to Brightspace named `A3_<studentnumber>.zip`, containing the required output file(s) listed above, your `code/` folder, and `reflection.md`. Run the format checker in `starter/` on your zip first — it validates file names, schemas, and value ranges locally before you submit. Auto-grading scores your output file against the hidden reference; the format checker passing is a precondition for grading.