

conflibertR :: cheat sheet

ConflibertR in R: inference, fine-tuning, model comparison, and active learning for conflict and political violence text.



Setup (once)

```
install.packages("conflibertR")
library(conflibertR)
conflibert_install() # Python backend
# restart R afterwards
```

Backend goes in a conflibert conda env (or virtualenv). Planning to use QA? Run `conflibert_install(qa = TRUE)` instead.

Health checks

```
conflibert_status() # full diagnostic
conflibert_available() # TRUE/FALSE guard
```

Inference (no training, no seed)

All functions are vectorized over character vectors and return tibbles that compose with `dplyr`. Results are deterministic.

Named Entity Recognition

```
ents <- conflibert_ner(
  "NATO forces were deployed near Kabul.")
# tibble: label, score, start, end
dplyr::filter(ents, label == "Location")
```

Binary classification (conflict / not)

```
conflibert_classify(
  c("A bomb exploded in the market.",
    "The weather was sunny and warm."))
# label, class, confidence, prob_*
```

Multilabel classification

```
conflibert_multilabel(texts)
# one probability column per label
```

Question answering

```
conflibert_qa(
  context = "Rebels seized the town in May.",
  question = "Who seized the town?")
conflibert_qa(context, question,
  details = TRUE) # + scores
```

Quick benchmark of the base classifier

```
conflibert_benchmark(texts, labels)
```

Returns accuracy, precision, recall, and F1 as a one-row tibble.

Bundled example data

```
data <- conflibert_example("binary")
# also "multiclass" and "active"
str(data) # $train / $dev / $test
```

Each element is a data.frame with text and label columns, the format every training function expects.

Fine-tune your own classifier

```
fit <- conflibert_finetune(
  train, dev, test,
  model = "Conflibert",
  task = "binary", # or "multiclass"
  epochs = 3, batch_size = 8, lr = 2e-5,
  save_dir = "my_model",
  seed = 42) # reproducible
fit$metrics # test-set tibble
fit$predictions; fit$probabilities
plot(fit) # confusion matrix
```

Low-memory training (LoRA)

```
conflibert_finetune(..., use_lora = TRUE,
  lora_rank = 8, lora_alpha = 16)
# ~5x less GPU memory; adapter is
# merged before saving
```

Reload and predict later

```
clf <- conflibert_load("my_model")
predict(clf, new_texts)
```

Compare base models

```
cmp <- conflibert_compare(
  train, dev, test,
  models = c("Conflibert",
    "BERT Base Uncased",
    "ModernBERT Base"))
cmp # metrics side by side
plot(cmp) # comparison chart
```

`conflibert_models()` lists all 7 choices: Conflibert, BERT Base (un)cased, RoBERTa Base, ModernBERT Base, DeBERTa v3 Base, DistilBERT Base.

Active learning

Label only the texts the model is least sure about: start from a small labeled seed set, iterate in rounds.

```
data <- conflibert_example("active")
s <- conflibert_active_start(
  seed = data$seed, # labeled df
  pool = data$pool, # unlabeled df
  dev = data$dev,
  query_size = 10,
  strategy = "entropy", # or "margin",
  # "least_confidence"
  diverse = FALSE, # TRUE = spread
  # batch via k-means
  random_seed = 42)
s$query # texts to label next
```

Label a round, retrain, repeat

```
# point-and-click Shiny gadget:
labels <- conflibert_active_label(s)
# ...or supply labels programmatically
s <- conflibert_active_next(s, labels)
s$metrics # per-round history
plot(s) # learning curve
conflibert_active_save(s, "al_model")
```

Plots & output

Training and inference objects have themed `print()` methods; results stay plain tibbles underneath.

```
plot(fit); plot(cmp); plot(s)
ggplot2::autoplot(fit) # same, as
# ggplot object
+ theme_conflibert() # package theme
```

Reproducibility

Training functions take `seed` (`conflibert_active_start()`: `random_seed`), default 42: same seed, hardware, and package versions give identical metrics and predictions. Change it to study run-to-run variability.

Citing

Cite Hu et al. (2022), *NAACL-HLT*, for the Conflibert model, and Brandt et al. (2025), *Political Analysis*, for its validation against generative LLMs. `citation("conflibertR")` gives BibTeX for both.