



Text Classification

NLP tasks 01 | Binary, multi-class, multi-label and hierarchical labelling of text: benchmark datasets, TF-IDF baselines to fine-tuned encoders, SetFit, zero-shot NLI and LLM prompting, label design, thresholds and cascades, confusion analysis, pitfalls in production

#NLP #Classification #SetFit #Cascade #MacroF1

● Definition and the entry point

Text classification

×1

Put each text in the right box: spam or not, which department, which topic, toxic or not.

×10

Choose the label scheme first: multi-class (one box), multi-label (several boxes), hierarchical (boxes in boxes); add an out-of-scope box.

- Score with macro-F1 and per-class recall
- Tune the decision threshold to error costs

×100

A model maps text to a vector (TF-IDF counts, encoder [CLS] or pooled embedding, or an LLM's reading) and a head turns it into class scores; calibration, thresholds and abstention turn scores into decisions.

- Encoder fine-tuning: cross-entropy on a linear head
- LLM: label names and definitions in the prompt, constrained to the label set

ANALOGY

A mail room: the sorter (model) is only as good as the tray labels (taxonomy) and the rule for 'not sure, send to supervisor' (threshold).

Before Text classification, can you explain. . .

		Not yet	Partly	Yes
MUST KNOW	Precision, recall, F1, confusion matrix	☐	☐	☐
Learn: machine-learning 07				
MUST KNOW	Logistic regression and softmax	☐	☐	☐
Learn: machine-learning 03				
MUST KNOW	TF-IDF features	☐	☐	☐
Learn: perception-and-generation 09				
HELPFUL	Calibration and imbalanced data	☐	☐	☐
Learn: machine-learning 09				
HELPFUL	Fine-tuning recipe (LR, epochs) for encoders	☐	☐	☐
Learn: losses-and-training 10				
ADVANCED	Cost-sensitive thresholds	☐	☐	☐
Learn: metrics-and-kpis 02				

Dataset	Task	Size and labels	Note
SST-2 / SST-5	movie-review sentence sentiment	≈ 67k train sentences (SST-2, GLUE); 5 classes in SST-5	part of GLUE; saturated
IMDb	review polarity	50k reviews, 25k train / 25k test	long texts, binary

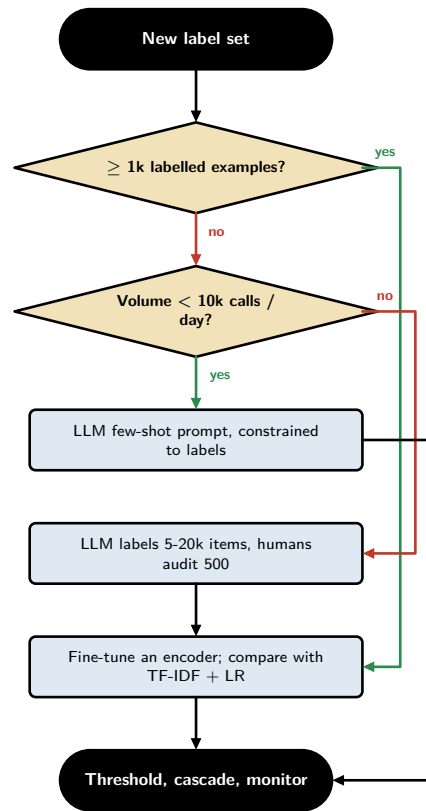
AG News	news topic	120k train, 7.6k test, 4 classes	classic topic benchmark
DBpedia-14	Wikipedia entity type	560k train, 14 classes	near-ceiling accuracy
20 Newsgroups	newsgroup topic	≈ 18.8k posts, 20 classes	headers leak the label: strip them
BANKING77	intent detection	≈ 13k banking queries, 77 intents	fine-grained, similar intents
CLINC150	intent + out-of-scope	150 intents in 10 domains + OOS queries	tests rejection
Reuters-21578, EUR-Lex	multi-label topics	Reuters ≈ 90 used labels; EUR-Lex thousands	long-tail labels
Jigsaw / Civil Comments	toxicity	hundreds of thousands of comments, several toxicity labels	identity-term bias studied here

Benchmark datasets you will meet (sizes approx.)

🔗 Approaches and trade-offs

Approach	Labelled data	Strengths	Weaknesses
Keyword rules, regex	none	transparent, instant, exact codes	brittle, low recall, endless rule upkeep
Naive Bayes / logistic regression on TF-IDF (1-2-grams)	hundreds to thousands	trains in seconds on CPU, strong baseline on topic tasks	no word order or paraphrase; weak on sentiment nuance
fastText (2016)	thousands	linear model on word + n-gram embeddings, very fast	same limits as bag of words
Fine-tuned encoder (BERT, RoBERTa, DeBERTa-v3, ModernBERT)	1k-10k	best accuracy per euro at scale, ms latency on GPU	needs labels, retraining for new classes
Embeddings + logistic regression	hundreds	one embedding model serves many classifiers	below fine-tuning on subtle classes
SetFit (few-shot)	8-64 per class	near fine-tuned quality from very few labels, small model	contrastive pairs grow quadratically; keep classes few
Zero-shot NLI (e.g. an MNLI-trained BART or DeBERTa)	none	label names as hypotheses, no training	one forward pass per label: slow with many labels
LLM zero / few-shot prompt	none to 20 examples	label definitions in plain words, new classes instantly, explanations	cost and latency per call, drift across model versions
LLM-labelled data → distilled encoder	LLM labels + 200-1,000 human checks	LLM quality at encoder cost	inherits LLM biases; needs audit

From rules to LLMs: what each approach needs and gives



Choosing an approach for a new classification task

⚠️ Constrain LLM labels

Free-text LLM answers drift ('Billing issue', 'billing', 'Payments'). Use an enum in a JSON schema or structured-output mode, give each label a one-line definition plus one borderline example, and add an explicit other label. Log the raw output to catch new classes users invent.

📊 Data, labelling and metrics

- **Taxonomy before data** ≤ 20 -30 top-level classes a human can hold in mind; mutually exclusive definitions for multi-class; an **other** class; examples on both sides of each boundary.
- **How much data** a few hundred examples per class for an encoder; 8-64 per class for SetFit; for LLM prompts, enough eval data to measure each class (≥ 30 -50 per class).
- **Agreement** double-label 100-200 items; classes with low kappa need clearer rules or merging (data 04 computes kappa).

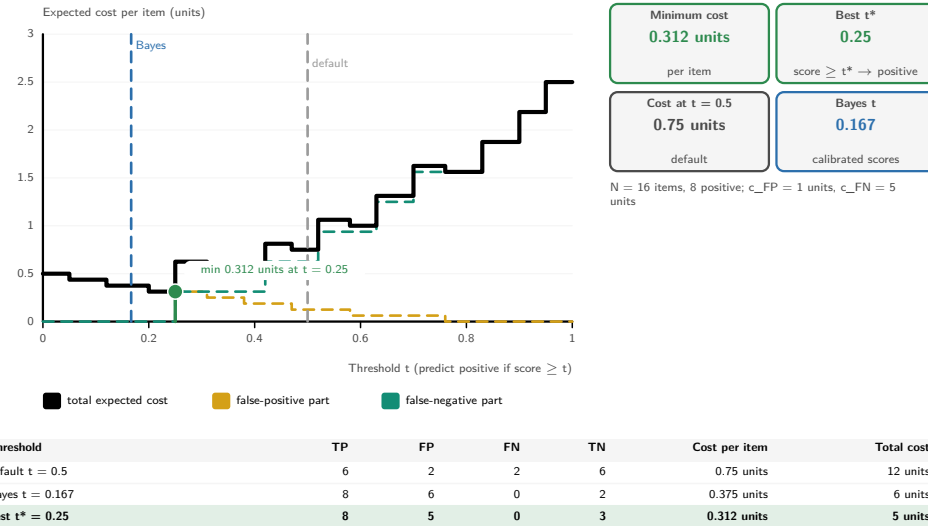
- **Splits** deduplicate (exact and near-duplicate hashing) before splitting; split by time or by customer when the model will see future or new customers.

Actual	Predicted			Total
	Billing	Refund	Technical	
Billing	90	25	5	120
Refund	20	40	0	60
Technical	5	5	110	120
Total	115	70	115	300
Precision	0.78	0.57	0.96	
Recall	0.75	0.67	0.92	
F1	0.77	0.62	0.94	

Accuracy 80% · n = 300

Support-ticket router on 300 test tickets: the matrix computes per-class precision, recall and F1. Refunds are confused with Billing, the signal to merge or redefine them

Metrics recall (formulas in machine-learning 07 and metrics-and-kpis 02): report **macro-F1** plus **per-class recall** for imbalanced sets; accuracy only when classes are balanced; **micro-F1**, **Hamming loss**, **subset accuracy** for multi-label; **PR-AUC** when the positive class is rare; **ECE** when scores drive decisions. For intent detection also report **OOS recall** (rejected out-of-scope queries).



Cost(t) = (c_{FP} · FP(t) + c_{FN} · FN(t) + c_{TP} · TP(t) + c_{TN} · TN(t)) / N; t* = the threshold of minimum Cost(t)
 Bayes threshold for calibrated probabilities: $t = (c_{FP} - c_{TN}) / ((c_{FP} - c_{TN}) + (c_{FN} - c_{TP})) = 1 / 6 = 0.167$
 where t = decision threshold (positive when score ≥ t); TP, FP, FN, TN = counts at t; c_{FP}, c_{FN}, c_{TP}, c_{TN} = cost of each outcome in units; N = items (16)

Urgent-ticket detector: a missed urgent ticket costs 5× a false alarm, so the cost-minimising threshold falls below 0.5

● Cascades in production

PROBLEM A fine-tuned encoder handles tickets it scores ≥ 0.9; the rest go to an LLM. Overall accuracy and LLM call share? (illustrative figures)

GIVEN

- Share of tickets with encoder confidence ≥ 0.9: c = 0.80, accuracy on them a_e = 0.96
- LLM accuracy on the remaining hard tickets a_l = 0.88
- Relative cost per ticket: encoder 1 unit, LLM 100 units

1 Accuracy of the cascade

$$A = c \times a_e + (1 - c) \times a_l = 0.8 \times 0.96 + 0.2 \times 0.88 = 0.768 + 0.176 = 0.944$$

Why: weighted by the traffic each path handles

2 Cost per ticket

every ticket pays the encoder: $1 + 0.2 \times 100 = 21$ units

Why: the encoder runs first on all traffic

3 LLM-only cost

100 units per ticket

Why: baseline to compare

4 Saving

$$1 - 21 / 100 = 0.79$$

Why: 79 % cheaper

ANSWER 94.4 % accuracy at 21 % of the LLM-only cost; move the 0.9 threshold to trade cost for accuracy

CHECK ✓ If the LLM alone reaches 95 %, the cascade loses 0.6 points: decide with the error cost, not accuracy alone

Calibrate first

temperature scaling on a held-out set so 0.9 means 90 % correct; raw softmax of fine-tuned encoders is over-confident

Per-class thresholds

multi-label models need one threshold per label, tuned on validation F1 or cost

New classes

an LLM handles a new label the same day; retrain the encoder once 200+ examples exist

Monitoring

predicted class distribution, mean confidence, share sent to LLM or human, weekly sample re-labelled

📖 EXAM TIP

Interview trap: 'Accuracy is 95 %, ship it?' Ask for the class distribution: if 95 % of tickets are Billing, a constant predictor scores 95 % with zero recall on everything else. Answer with macro-F1, per-class recall and the cost of each error type.

🚩 Pitfalls, myths and recap

Pitfall	Symptom	Fix
Leakage via duplicates or headers	test score far above production	near-duplicate removal, strip signatures and headers, group-aware split
Spurious cues	model keys on a token ('urgent!!!', a product code)	error analysis, counterfactual edits, more diverse data

Label drift	accuracy decays as products and policies change	time-based eval, monthly sample, retrain trigger
Truncation	long emails classified on the greeting	use the first and last 256 tokens, or a long-context encoder (ModernBERT: 8,192 tokens)
Identity-term bias	toxicity flagged for 'gay', 'Muslim' in neutral text	per-group FPR (responsible-ai 03), balanced counterexamples
Prompt sensitivity	LLM label shifts with example order or model update	fixed eval set run on every prompt or model change

Pitfalls seen in real projects

MYTH OR FACT?
"With an LLM you no longer need labelled data"

MYTH: You still need labelled eval data
Without 200+ labelled examples you cannot tell a 70 % prompt from a 90 % one.

MYTH OR FACT?
"Bag-of-words baselines are pointless now"

MYTH: TF-IDF + LR often lands within a few points on topic tasks
It trains in seconds and reveals leakage when it scores suspiciously high.

MYTH OR FACT?
"Softmax 0.99 means the model is sure"

MYTH: Fine-tuned networks are often over-confident
Calibrate before thresholding or cascading.

MYTH OR FACT?
"Multi-label is just multi-class with more labels"

MYTH: It needs sigmoid outputs and per-label thresholds
Softmax forces labels to compete; independent labels should not.

Tick a box only when you could explain the point to someone else

- ☐ Write the taxonomy with definitions and an other class
agreement first
- ☐ Fine-tune an encoder at volume; SetFit with few labels
- ☐ Calibrate, then set thresholds from error costs
- ☐ Baselines: TF-IDF + LR and a constrained LLM prompt
- ☐ Macro-F1 and per-class recall, not accuracy
- ☐ Cascade: small model first, LLM or human for the uncertain

Q1 Output layer and loss for multi-label classification?

A One sigmoid per label with binary cross-entropy, and a threshold per label.

Q2 What does SetFit fine-tune, and with what?

A A sentence-transformer, contrastively on pairs built from a few labelled examples, then a logistic-regression head.

Q3 How does zero-shot NLI classification score a label?

A As the entailment probability of a hypothesis such as 'This text is about {label}.' given the text.

Q4 Why can 20 Newsgroups scores be misleading?

A Headers, signatures and quoted replies leak the newsgroup; strip them.

Q5 Cascade accuracy with 80 % of traffic at 0.96 and 20 % at 0.88?

A $0.8 \times 0.96 + 0.2 \times 0.88 = 0.944$.

Q6 Why calibrate before thresholding?

A So the score means a probability; raw softmax of fine-tuned models is over-confident.

Q7 Which metric for an intent detector with an out-of-scope class?

A Macro-F1 over intents plus out-of-scope recall.

Q8 How to stop an LLM inventing labels?

A An enum in a JSON schema or structured output, label definitions, an other label, and logging raw outputs.

Q9 When does a fine-tuned encoder beat an LLM prompt?

A Fixed label set, thousands of examples, high volume or tight latency.