Multilingual Social Engineering and Fraud Tagging Corpus*

Darwin Zhang **Marco Wang** **Tianhao Cao** **Yusen Huang**

Abstract

Phishing and spam emails cause major losses, but most classifiers assume monolingual data and large labeled datasets, assumptions that fail in bilingual (English/Chinese) settings with limited annotations. We study three-class email classification under an extreme low-resource setup. We build traditional (TF-IDF + LinearSVC) and neural (DistilBERT) baselines, then add cross-lingual transfer, ensembling, multi-task learning, and data expansion. Our best model, a grid-searched ensemble of TF-IDF+SVC and DistilBERT, achieves 0.8778 macro-F1, outperforming the neural baseline (0.5444), which fails on Spam. Ablations show that gains come mainly from subword robustness, Chinese-only embeddings, and ensembling, while NER hurts performance. Overall, in tiny bilingual settings, classical methods with strong ensembling and targeted augmentation outperform transformer fine-tuning, with ablation playing a key role.

1 Introduction

Social engineering fraud, phishing, spam, and deceptive solicitation, remains a major threat in digital communication. Rather than exploiting software flaws, these attacks target human psychology through urgency, authority, fear, and reward to steal credentials, money, and personal information. English-only detection systems are increasingly insufficient as fraud adapts across languages, platforms, and styles.

To address this gap, we present **FRAUD** (Fast Repository for Analyzing Unreliable Discourse)[1], a multilingual annotated corpus for fraud detection beyond bag-of-words, English-only baselines. Built on a large-scale English email dataset, FRAUD adds bilingual Mandarin Chinese annotations and fine-grained document- and span-level labels that capture not only whether fraud occurs, but how it is conveyed. The corpus uses three document labels, Ham, Phish, and Spam, plus two annotation layers: (1) a `tactic_primary` field marking the dominant psychological strategy (AUTHORITY, URGENCY, THREAT, REWARD, HELPFUL_SERVICE), and (2) span-level NER over entities commonly exploited in fraud (persons, organisations, URLs, credentials, monetary amounts, system identifiers). Each English email is paired with a machine-translated Mandarin counterpart sharing the same label, enabling cross-lingual transfer.

Analysis of the 72.6M-token base dataset reveals class-distinctive patterns: Ham messages are longest and least lexically diverse; Phish messages are shortest and most diverse; Spam lies between and often includes obfuscation artefacts designed to evade filters. These patterns motivate both the classification task and our modelling choices.

Beyond the corpus, we conduct a systematic study of what improves extreme low-resource bilingual fraud detection. Starting from a TF-IDF+LinearSVC baseline and a fine-tuned DistilBERT baseline, we add (a) cross-lingual transfer via frozen fastText CC-100 vectors and mDistilBERT, (b) a bias-corrected, entropy-weighted, grid-searched ensemble, (c) multi-task learning with silver NER supervision, and (d) data expansion via bootstrapping, active learning, and LLM-generated few-shot examples. Four ablations reveal that gains stem mainly from subword robustness and ensemble design, while auxiliary NER hurts performance at this scale.

2 Methodology

2.1 Baselines

We pair two complementary baselines. B1 (Traditional): TF-IDF (uni- and bigrams) + LinearSVC with a bilingual tokenizer (Jieba for Chinese, regex for English) and class weighting. B2 (Neural): distilbert-base-uncased fine-tuned on the English text field with class-weighted cross-entropy, AdamW (lr = 2e-5), 10% linear warmup, and early stopping on validation macro-F1. B1 covers both languages but ignores word order; B2 captures contextual semantics but is monolingual and data-hungry — together they bracket the design space for later experiments.

*Repository

[1]Project name acronym.

2.2 Transfer Learning

To address each baseline's weakness, we introduce two transfer variants. T1 (Traditional + fastText): the TF-IDF feature vector is concatenated with mean-pooled, frozen fastText CC-100 vectors covering both English and Chinese (1525-d total), supplying subword robustness and cross-lingual coverage to the SVC. T2 (Multilingual neural): the DistilBERT backbone is swapped for distilbert-base-multilingual-cased, with the bottom three of six transformer layers frozen to mitigate catastrophic forgetting on a 105-example training set.

2.3 Ensembling

We combine B2 and B1 by weighted soft voting:

$$\hat{y} = \arg\max_c \left(w_{B1}\, p_c^{B1} + w_{B2}\, p_c^{B2} \right),$$

$$\text{where } w_{B1} + w_{B2} = 1.$$

LinearSVC decision scores are converted to probability-like values via softmax.

E1 (Naive ensemble) uses fixed weights.

E2 (Motivated ensemble) adds three components: (a) per-class **bias correction**, which rescales each model's class probabilities using its validation precision profile; (b) **entropy-based down-weighting**, which reduces the contribution of high-uncertainty predictions via Shannon entropy; (c) a **grid search** over 21 weight candidates on the validation set.

2.4 Multi-Task Learning with Silver NER

We hypothesise that fraud is partly characterised by entity usage—Phish impersonates ORG/MONEY/DATE; Spam is entity-light; Ham has diffuse entity distributions. Silver NER labels are generated automatically by `spaCy en_core_web_sm`.

MTL-Neural. A single `distilbert-base-uncased` encoder feeds two heads: a classification head (`nn.Linear(H, 3)`) over the `[CLS]` pooled vector, and a NER head (`nn.Linear(H, 21)`) over every wordpiece's hidden state, producing per-token BIO logits across 21 entity labels. Silver labels are aligned to WordPiece tokens via offset mapping. The joint loss is

$$L = L_{\text{cls}} + \lambda L_{\text{ner}}, \quad \lambda = 0.3,$$

so gradients from both objectives flow into the shared encoder; only the classification head is used at inference.

MTL-Traditional. We compute per-document densities (proportion of tokens) for five entity types—ORG, MONEY, DATE, PERSON, GPE—using `spaCy en_core_web_sm`. These five dense features are concatenated with the sparse TF-IDF matrix before training `LinearSVC`.

2.5 Data Augmentation

We expand the 105-sample gold training set with three pipelines on top of a 1,104-sample base (105 gold + 999 silver-NER).

Bootstrapping. The best Sprint-2 model (TF-IDF + LinearSVC) pseudo-labels 26 unannotated texts (25% of the gold set), which are appended to the training set (1,130 total). A fresh LinearSVC is retrained from scratch to avoid leakage.

Active learning. The 26 samples are re-ranked by (i) *margin sampling*—the gap between top-1 and top-2 `decision_function` scores—and (ii) *vote entropy* across a 5-model committee, each trained on a 90% subsample. Pseudo-labels are then replaced with gold labels, and accumulation tests retrain LinearSVC on the base plus top-N samples ($N \in \{2, 3, 4, 6, 26\}$).

Few-shot LLM augmentation. Guided by AL error analysis (Ham→Phish/Spam, Phish→Spam), we prompt an LLM with three in-context exemplars to generate 21 boundary cases (7 per class) in the corpus JSON schema. Combined with two retained drafts, this yields a final 1,153-sample dataset.

2.6 Ablations

We run four ablations to isolate which design decisions actually drive performance across Sprints 1–3. **Ablation 1** disentangles whether T1's gain comes from fastText's semantic geometry or its subword coverage by replacing fastText with character-level TF-IDF (`analyzer='char_wb'`, n=$(2, 4)$), which provides subword patterns without semantic structure. **Ablation 2** sweeps the Neural MTL auxiliary loss weight $\lambda \in \{0.0, 0.3, 1.0\}$, with λ=0.0 as a control in which the NER head runs but contributes no gradient, testing whether the Sprint-3 setting is optimal. **Ablation 3** removes each of E2's three layers in turn—per-class bias correction, per-sample entropy weighting, and grid-searched model weights—to attribute the ensemble's gains to specific components. **Ablation 4** varies fastText language coverage (EN-only, ZH-only, EN+ZH) to

test whether each language's vectors carry independent signal. All four are evaluated on the standard 12-sample validation and 16-sample test splits.

3 Data

3.1 Source and Corpus Construction

Our corpus is derived from *The Biggest Spam Ham Phish Email Dataset*[2], a large-scale English email collection with over 300,000 samples labelled as **Ham**, **Phish**, or **Spam**. From this base, we curated a gold-annotated subset and added machine-translated Mandarin Chinese counterparts, producing a bilingual corpus in which each English email is paired with a `text_zh` field.

3.2 Annotation Schema

Each record includes three additional annotation layers.

Document-level tactic Phish and Spam emails are assigned a `tactic_primary` label indicating the main psychological strategy: `AUTHORITY`, `URGENCY`, `THREAT`, `REWARD`, or `HELPFUL_SERVICE`. Ham records may take `null`.

Scenario Each email is assigned a scenario categories, such as `IT_SYSTEM`, `LEGAL_REGULATORY`, `FRAUD_SOCIAL_ENGINEERING`, or `HR_RECRUITING`.

Span-level NER Entities are annotated using character offsets on masked text, covering core types such as `PERSON`, `ORG`, `DATE`, `MONEY`, `URL`, `EMAIL`, `PHONE`, and `CREDENTIAL`, along with task-specific subtypes.

3.3 Preprocessing and Splits

Duplicate records were removed by exact text matching to avoid leakage. The deduplicated corpus was then split into train, validation, and test sets using a stratified 80/10/10 split with fixed seed 581. A post-split check confirmed zero text overlap across splits.

The final gold corpus contains **133 records**: 105 train, 12 validation, and 16 test.

4 Experimentation & Results

Our experiments follow a progressive design: each stage targets a weakness identified in the previous one. We use a 105/12/16 train/val/test split (seed 581) and report accuracy and macro-F1.

[2]Data Source

Split	Total	Ham	Phish	Spam
Train	105	36	40	29
Validation	12	3	5	4
Test	16	5	6	5
Total	**133**	**44**	**51**	**38**

Table 1: Class distribution of the gold-annotated corpus across train, validation, and test splits.

4.1 Baselines (B1, B2)

B2 (TF-IDF + LinearSVC) achieves 0.6250 test accuracy and 0.6405 macro-F1, detecting all classes but confusing Spam with Phish (Table 2). B1 (DistilBERT) reaches 0.6875 / 0.5444—higher accuracy but a complete Spam blindspot (F1 = 0.00, 0/5 recall; Table 3). This reflects class imbalance under large model capacity.

True \ Pred	Ham	Phish	Spam
Ham	3	0	0
Phish	0	5	0
Spam	0	2	2

Table 2: B2 (TF-IDF + SVC) confusion matrix

True \ Pred	Ham	Phish	Spam
Ham	5	0	0
Phish	0	6	0
Spam	2	3	0

Table 3: B1 (DistilBERT) confusion matrix

4.2 Transfer Learning (T1, T2)

T1 (TF-IDF + fastText) improves macro-F1 from 0.6405 to 0.7566, with gains in Phish and non-zero Spam F1. T2 (mDistilBERT, bottom-3 frozen) raises validation macro-F1 to 0.8110 and partially recovers Spam (F1 $\approx$ 0.25), but test macro-F1 drops to 0.4101, indicating poor generalisation under limited data. Validation is the more reliable signal given the small test set.

4.3 Ensembling (E1, E2)

E1 (naive soft voting, 0.3 / 0.7) achieves 0.6349 macro-F1, similar to B2, as DistilBERT's miscalibrated Spam probabilities degrade performance. E2 combines bias correction, entropy weighting, and grid-searched weights, reaching 0.8750 accuracy and 0.8778 macro-F1. Grid search selects $w_{\text{bert}} = 0.00$ post-correction, effectively choosing bias-corrected SVC (see §2.6).

4.4 Multi-Task Learning (Silver NER)

Neural MTL (DistilBERT + NER head, $\lambda = 0.3$) improves validation macro-F1 ($0.5714 \rightarrow 0.7231$) and speeds convergence, but fails on test (Spam F1 = 0.00). With only 5 Spam samples, results are highly sensitive; the validation gain is more informative.

Traditional MTL (TF-IDF + entity-density features) degrades validation performance ($0.9000 \rightarrow 0.8300$), suggesting feature mismatch in low-data settings.

Configuration	Val F1	Test Acc	Test F1
B1 DistilBERT	0.5714	0.6875	0.5444
T2 mDistilBERT	0.8110	0.4375	0.4101
Neural MTL ($\lambda = 0.3$)	0.7231	0.5625	0.4353
B2 TF-IDF + SVC	0.9000	0.6250	0.6405
Traditional MTL	0.8300	—	—

Table 4: MTL vs. baseline comparisons

4.5 Data Augmentation

We expand the training set using three methods:

Bootstrapping. Pseudo-labelling 26 samples yields marginal improvement (dev accuracy 0.9167), suggesting saturation.

Active learning. SVC margin sampling identifies plausible boundary cases, but a 5-model committee shows zero disagreement (entropy = 0), indicating limited diversity under small data.

Few-shot LLM augmentation. Generating 21 hard-boundary samples yields the largest gain, improving dev accuracy from 0.9167 to 1.0000.

4.6 Ablations

Ablation 1 — Subword robustness. T1 improved test macro-F1 from 0.6405 to 0.7566, but fastText provides both semantic geometry and subword coverage. To disentangle these, we replace fastText with character-level TF-IDF (`analyzer='char_wb'`, $n = (2, 4)$).

Configuration	Test Acc	Test F1
Word TF-IDF only	0.6875	0.7014
Word + Char TF-IDF	0.7500	0.7556
Word TF-IDF + fastText	0.7500	0.7566

Character TF-IDF nearly matches fastText ($\Delta = 0.001$), indicating gains come from subword coverage rather than semantic geometry.

Ablation 2 — NER loss weight λ. We sweep $\lambda \in \{0.0, 0.3, 1.0\}$ in Neural MTL, where $\lambda = 0.0$ is a control (no gradient from NER).

λ	Best Val F1	Test Acc	Test F1
0.0	0.8110	0.6875	0.6909
0.3	0.7231	0.5625	0.4353
1.0	0.7231	0.5625	0.4444

$\lambda = 0.0$ performs best and is the only setting with non-zero Spam F1. NER supervision degrades performance, likely due to unstable optimisation and a small validation set.

Ablation 3 — Ensemble components. E2 combines bias correction, entropy weighting, and grid-searched model weights. We remove each component:

Configuration	Test Acc	Test F1
Full ensemble	0.8750	0.8778
No bias correction	0.6250	0.6405
No entropy weighting	0.8750	0.8778
No grid search	0.5625	0.4627

Grid search is critical, selecting $w_{\text{bert}} = 0.00$ (SVC-only). Bias correction is also essential, while entropy weighting is redundant.

Ablation 4 — fastText language coverage. We test EN vs. ZH fastText embeddings:

Configuration	Test Acc	Test F1
TF-IDF only	0.6875	0.7014
TF-IDF + EN fastText	0.7500	0.7381
TF-IDF + ZH fastText	0.8750	0.8778
TF-IDF + EN+ZH fastText	0.7500	0.7566

ZH-only unexpectedly performs best (0.8778), likely due to near-zero embeddings acting as a regulariser. However, identical validation F1 (0.8110) across settings suggests this variance should not be over-interpreted.

5 Conclusion

We presented FRAUD, a bilingual (English/Chinese) email corpus annotated for fraud detection at the document, tactic, and span level, and a systematic study of what actually helps in extreme low-resource bilingual classification. From two complementary baselines, TF-IDF + LinearSVC and fine-tuned DistilBERT, we layered cross-lingual transfer, ensembling, multi-task learning with silver NER, and three data-expansion pipelines, and ran four targeted ablations to attribute every gain.

Three findings are robust across our experiments. First, **classical pipelines outperform transformer fine-tuning** at this scale: our best system is a bias-corrected, grid-searched ensemble that effectively reduces to TF-IDF + LinearSVC ($w_{\text{bert}} = 0$), reaching **0.8778 macro-F1**. This result is 37 absolute

points above the DistilBERT baseline, which never predicts Spam. Second, **gains attributed to modern components often come from elsewhere**: fastText's lift is driven almost entirely by subword robustness rather than semantic geometry, and ZH-only embeddings match the full ensemble plausibly through a regularising near-zero-vector effect. Third, **auxiliary supervision can hurt**: NER multi-task learning, motivated by clear class-conditional entity differences, degrades both neural and traditional models because the joint loss destabilises checkpoint selection on a 12-sample validation set.

For data expansion, bootstrapping and active learning saturate quickly on a 1,104-sample base, while LLM-generated boundary cases targeting concrete error modes (Ham $\rightarrow$ Phish/Spam, Phish $\rightarrow$ Spam) are the only intervention to push dev accuracy from 0.9167 to 1.0000. The lesson is that augmentation is most useful when shaped by error analysis rather than applied generically.

Our results are bounded by a 16-sample test set, which makes single-misclassification swings of ± 0.04 macro-F1 unavoidable. We therefore lean on validation trends and ablation deltas as the load-bearing evidence. The broader takeaway is methodological: in low-resource bilingual fraud detection, careful ablation, not architectural sophistication, is the deciding lens, and disciplined ensembling combined with targeted synthetic augmentation outperforms end-to-end transformer fine-tuning. Building upon our initial objective to facilitate the learning of additional languages using only a small subset of data through transfer learning, future work will explore expanding these lightweight, highly tuned pipelines to new linguistic domains. By leveraging cross-lingual embeddings and targeted synthetic data expansion, we aim to demonstrate that fraud detection models can scale across languages without the prerequisite of massive annotated datasets.

Generative AI Use Disclosure

Generative AI tools were used during the preparation of this work for several purposes: assisting with code generation in areas where the authors lacked sufficient expertise, such as front-end development; improving the robustness of existing code through suggestions on error handling, pipeline structure, and edge case coverage; and grammar checking and light proofreading of the written manuscript. All generated code was manually reviewed by the authors to verify correctness and ensure it behaved as intended. All scientific content, experimental decisions, analysis, and conclusions are entirely our own.