Gender Classification of Blog Authors Using Contrastive Learning

Mahtab Jeyhani and **Minh Nguyen**
University of Houston

Abstract

Predicting an author's gender from their writing is a challenging NLP task: Differences in male versus female style are subtle and individual variation is high. Traditional feature-based methods struggle to generalize, and fine-tuning large pre-trained models like BERT can overfit when labeled data are scarce. In this work, we employ supervised contrastive learning to learn gender-discriminative text representations. By explicitly pulling together examples of the same gender and pushing apart those of different genders during pre-training on a large labeled blog corpus (600K+ posts), our encoder learns robust embeddings that capture stylistic patterns beyond simple topic cues. We then fine-tune a lightweight classifier in our smaller 3.2K post gender-labeled data set. Experiments show that our two-stage pipeline produces more stable training dynamics and performs better than traditional baselines, such as logistic regression and SVM with TF-IDF and bag-of-words features, in overall classification accuracy.

1 Data Preparation

1.1 Dataset Description

External Blog Corpus (J. Schler, M. Koppel, S. Argamon and J. Pennebaker, 2006) For contrastive pre-training, we leverage a large external blog dataset comprising over 600,000 posts authored by a diverse set of bloggers. This corpus is roughly balanced between genders (51% male, 49% female) and spans a wide range of topics, from personal journals to technical tutorials, ensuring that our encoder sees varied writing styles and lexical patterns. On average, posts contain 200–250 words, although some entries exceed 3,000 words. This breadth of length and content helps the model learn general stylistic representations that extend beyond topic-specific signals. This dataset is used to prevent any data leakage later in the fine-tuning phase.

Gender-Labeled Dataset (Mukherjee and Liu, 2010) For supervised fine-tuning and evaluation, we use the publicly available blog authorship dataset of Mukherjee and Liu (2010), which contains 3,226 posts labeled by author gender. The split is approximately 52% male and 48% female. Each example is a free-form blog entry, with an average length of 222 words per post. This variability in length, topic, and personal style makes gender signals subtle, motivating our use of robust, contrastively pre-trained embeddings before task-specific fine-tuning.

1.2 Data Preprocessing

Before any modeling, we apply the following standard cleaning steps to each blog post:

- **Lowercasing:** Convert all text to lowercase to reduce vocabulary size.
- **HTML Unescaping:** Replace HTML entities (e.g., ` `) with their Unicode equivalents.
- **URL Removal:** Strip out any URLs via regex (e.g., `http://...`).
- **Punctuation Removal:** Drop punctuation characters to focus on word tokens.
- **Stopword Removal:** Remove common English stopwords (NLTK list) to emphasize content words.
- **Whitespace Normalization:** Collapse multiple spaces into one and trim leading/trailing spaces.

- **Label Normalization:** Normalize the gender labels by removing spaces, convert labels to binary values (0 for Male and 1 for Female).
- **Drop NA samples:** Drop samples with null text or gender value.

These steps help reduce noise and standardize inputs before feeding them into BERT and the contrastive pipeline.

1.3 Exploratory Analysis

We ran POS-tagging on the cleaned text to inspect stylistic markers:

- **Sample Nouns:** `['time', 'see', 'i', 'couple', 'times', 'path', 'btw', 'mlt']`
- **Sample Verbs:** `['was', 'rewriting', 'scratch', 'scratch', 'uses', 'sampling', 'help', 'tracking']`
- **Sample Adjectives:** `['long', 'poor', 'difficult', 'metropolis', 'fresh', 'standard', 'easy']`
- **Sample Adverbs:** `['always', 'still', 'very', 'now', 'especially', 'too', 'actually']`

This POS tagging process helped in understanding stylistic and grammatical usage patterns that might serve as implicit indicators of gender in writing.

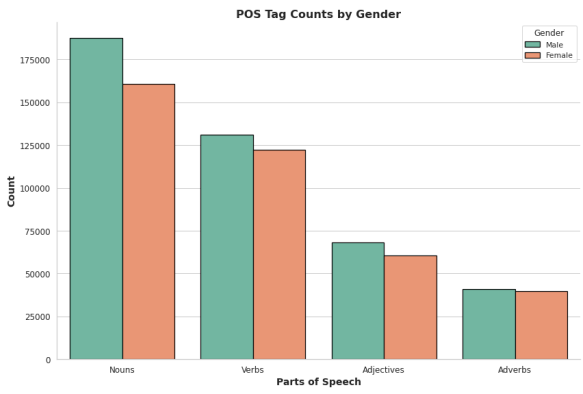

Figure 1: POS Tag Counts by Gender

Figure 1 illustrates the distribution of part-of-speech (POS) tag counts—nouns, verbs, adjectives, and adverbs—across male and female authors. While both groups follow a similar pattern in POS usage, male users consistently show higher counts in each category, with the largest disparity seen in noun usage. This is interesting because usually women are known for writing longer texts in general. This suggests potential gender-based differences in linguistic style or content focus.

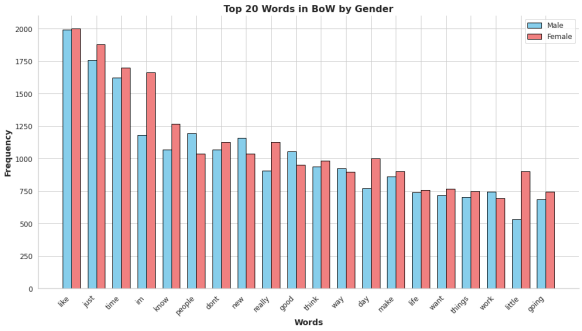

Figure 2: Top 20 Words in BoW by Gender

This bar chart (Figure 2) shows the top 20 most frequently used words by male and female participants based on a Bag-of-Words representation. While many of the words are common across both groups, subtle differences in frequency suggest potential variations in expression style. Words like “just”, “time”, and “know” appear more frequently among female users, whereas male users show slightly higher usage of words like “really”, “new”, and “people”. These patterns may reflect gender-based preferences in word choice or conversational context.

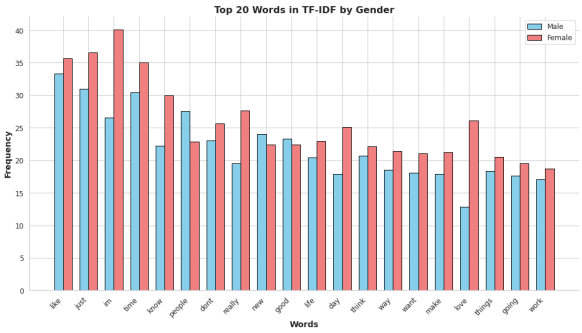

Figure 3: POS Tag Counts by Gender

Figure 3 presents the 20 most prominent words based on Term Frequency-Inverse Document Frequency (TF-IDF) values for male and female authors. TF-IDF helps highlight terms that are particularly distinctive within each group. While many of the top words overlap, female users generally exhibit higher TF-IDF scores, suggesting greater uniqueness or emphasis on certain terms

like “time”, “just”, and “love”. These differences may point to nuanced gender-specific communication patterns in language use.

1.4 Train/Val/Test Split

We then split the 3,226-post gender dataset into 64% train (2,065 posts), 16% validation (516), and 20% test (645) using stratified sampling to preserve the 52/48 male/female ratio.

2 Baseline Models

Before applying deep learning approaches, we established several traditional baselines using standard feature engineering techniques:

2.1 Feature Engineering

- **Bag-of-Words (BOW):** A sparse vector representation where each dimension corresponds to a vocabulary word, and entries reflect word counts.
- **TF–IDF:** A normalized weighting that reduces the influence of commonly occurring words and highlights more informative terms.

2.2 Classifier Training

Using these feature representations, we trained two types of baseline classifiers:

- **Logistic Regression**
- **Support Vector Machine (SVM)**

Both were evaluated via 5-fold cross-validation on the training set; the best hyperparameters were selected based on mean CV accuracy, and final performance was reported on the held-out test set.

Model	5-fold CV Acc.	Test Acc.
Logistic Regression + BOW	0.659 ± 0.012	0.655
Logistic Regression + TF–IDF	0.698 ± 0.020	0.695
SVM + BOW	0.629 ± 0.011	0.628
SVM + TF–IDF	0.683 ± 0.021	0.680

Table 1: Baseline classification results on the gender-labeled dataset.

Table 1 reports our baseline results. Among the feature-based methods, Logistic Regression with TF–IDF representations achieved the highest performance, with a 5-fold CV accuracy of 0.698 ± 0.020 and a held-out test accuracy of 0.695. Both Bag-of-Words variants and the SVM models lagged slightly behind—SVM+TF–IDF being the next best with 0.683 ± 0.021 CV and 0.680 test accuracy. These results confirm that classical linear classifiers over TF–IDF features remain strong baselines for this task. In the next section, we move on to our two-stage deep learning pipeline—combining supervised contrastive pre-training with a lightweight classifier—to see if we can surpass these traditional approaches.

3 Related Work

Self-supervised and contrastive learning have seen wide adoption across domains. In computer vision, SimCLR (Chen et al., 2020) laid the groundwork for simple contrastive frameworks, and recent NLP work has extended these ideas to text (Qian et al., 2022). For small data regimes, combining contrastive with masked autoencoder objectives has proven effective in medical imaging (Wolf et al., 2023). Supervised contrastive learning in NLP was formalized by Khosla et al. (Khosla et al., 2020), and BERT itself remains the language encoder (Devlin et al., 2019).

4 Methodology

4.1 Model Architecture

Our model, `BertContrastiveModel`, builds on top of a pre-trained BERT encoder (`bert-base-uncased`) (Devlin et al., 2019). It consists of:

- **Encoder:** BERT provides a pooled [CLS] embedding of size H (768).
- **Projection Head:** A two-layer MLP mapping the PCA-reduced inputs (64 dims) into the contrastive space:

$$h = \mathrm{ReLU}(W_1\, p + b_1), \quad v = W_2\, h + b_2,$$

  where $p \in R^{64}$, $h \in R^{H}$, and $v \in R^{d_{proj}}$ (e.g. $d_{proj} = 64$).
- **Classifier Head:** A simple feed-forward layer with dropout and a linear layer mapping $R^H \to R^2$ for final gender logits:

$$y = W_c\, z + b_c, \quad y \in R^2.$$

4.2 Supervised Contrastive Pre-training

4.2.1 Unsupervised Contrastive Learning

Initially, we experimented with unsupervised contrastive learning approach (Chen et al., 2020). We

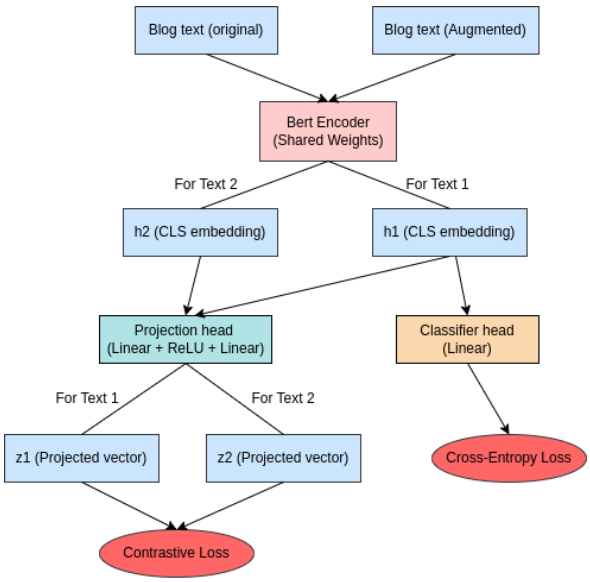

Figure 4: Overview of our BertContrastiveModel. Inputs (original and augmented) are tokenized and passed through BERT; pooled outputs are reduced via PCA, then projected into a contrastive space. A separate classifier head is used at fine-tuning time.

applied random text augmentations (random word dropout with a rate of 15% to 30%) and treated augmented pairs from the same input as positives. However, this approach yielded unstable training and suboptimal representations due to the noisy nature of augmentations in free-form blog text.

4.2.2 Supervised Contrastive Learning

To address this, we adopted supervised contrastive learning (Khosla et al., 2020), leveraging the external blog corpus which includes author gender labels. In this setup, each mini-batch was constructed such that all samples with the same gender formed positive pairs, and samples from different genders were treated as negatives. This allowed the model to learn more discriminative and robust representations by clustering embeddings of same-gender posts closer in the latent space while pushing apart those of different genders.

We also integrated PCA into the pre-training pipeline to reduce the dimensionality of the pooled BERT embeddings ($768 \rightarrow 64$), improving stability and efficiency. These reduced embeddings were then passed through the MLP projection head.

To efficiently leverage the large external blog corpus while conserving compute, we adopted the following pipeline:

1. **Text Preprocessing:** We applied the same cleaning steps (lowercasing, HTML unescaping, URL removal, punctuation, and stopword removal) to the external dataset before encoding.

2. **10K-Sample Subset:** Rather than pre-training on all 600K+ posts, we randomly sampled a balanced subset of 10,000 examples (5,000 male, 5,000 female). This dramatically reduced the training time. However, we believed that with powerful computer resources, training with the full corpus should yield even stronger representations.

3. **Data Augmentation:** To introduce additional variability, we applied random text augmentations—dropout of 15% to 30% of words for each sample to produce two "views" per input. This doubles the total size of the pre-training data.

4. **PCA Integration:** Each batch of pooled BERT embeddings ($z \in R^{768}$) was transformed via an offline-fitted PCA ($768 \rightarrow 64$) to produce $p \in R^{64}$. These reduced-dimension vectors served as inputs to the MLP projection head.

5. **Contrastive Objective:** We optimized the supervised contrastive loss (Khosla et al., 2020), which for a batch of N samples is given by:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\sum_{j:\,y_j=y_i}\exp(sim(v_i,v_j)/\tau)}{\sum_{k\neq i}\exp(sim(v_i,v_k)/\tau)},$$

where v_i is the projection-head output for sample i, τ is the temperature, and $sim(a,b) = a^\top b$ after ℓ_2 normalization.

This supervised setup pairs all same-gender examples as positives, encourages the encoder to cluster gender-specific writing styles tightly while separating opposite-gender styles in the learned embedding space.

4.2.3 Training Configuration

We pre-trained on the 10K-sample subset for 5 epochs with batch size 16 using Adam (learning rate $2\text{e}{-5}$, weight decay $1\text{e}{-4}$). Temperature τ was set to 0.2. PCA was applied on the CPU per-batch, and all gradient updates were back-propagated through the projection head only

(BERT was frozen). Training took approximately 1 hour over 5 epochs on a Mac M1 Pro GPU (MPS).

```
Supervised Contrastive Pre-training Epoch [1/5] Loss: 0.5511
Supervised Contrastive Pre-training Epoch [2/5] Loss: 0.4727
Supervised Contrastive Pre-training Epoch [3/5] Loss: 0.4484
Supervised Contrastive Pre-training Epoch [4/5] Loss: 0.4262
Supervised Contrastive Pre-training Epoch [5/5] Loss: 0.4084
Model saved at models/bert_supervised_contrastive_pretrained_final_pca.pth
Supervised Contrastive pre-training complete.
Supervised Contrastive pre-training complete. Model saved.
```

Figure 5: Supervised Contrastive Pre-training results over 5 epochs.

We observe a consistent decline in the contrastive loss from 0.55 down to 0.41 over five epochs, indicating that the model is indeed learning to cluster same-gender examples more tightly in the projection space while pushing apart opposite-gender examples. This downward trend confirms that the supervised contrastive objective is effectively shaping the embeddings before fine-tuning.

4.3 Supervised Fine-Tuning

After contrastive pre-training, we fine-tune the encoder's pooled representations on the smaller, gender-labeled dataset (3,226 posts). Our fine-tuning pipeline proceeds as follows:

1. **Train/Val/Test Split:** We stratify the 3,226-post gender dataset into 64% train (2,065 posts), 16% validation (516 posts), and 20% test (645 posts), preserving the 52% male / 48% female ratio.

2. **Classifier Setup:** We attach the pretrained BERT encoder (with frozen projection head) to the lightweight classifier head (dropout + linear layer mapping 768-dim pooled outputs to two logits).

3. **Optimization:** We train with AdamW (learning rate $2e{-5}$, weight decay $1e{-4}$) for up to 10 epochs, batch size 8. A cosine-with-warmup scheduler is used: 10% of total steps for linear warm-up, followed by cosine decay with half-cycle ($num\_cycles = 0.5$).

4. **Early Stopping & Checkpointing:** We monitor validation accuracy each epoch and save the best model. Training stops if validation accuracy does not improve for 3 consecutive epochs or if validation loss exceeds 0.9.

5. **Evaluation Metrics:** We report final test accuracy, precision, recall, and F1 score, as well as the full confusion matrix. Learning curves (train/val loss and accuracy over epochs) are also plotted to verify convergence and detect any overfitting.

5 Results

5.1 Learning Curves

Figure 6: Fine-tuning Train/Val Loss Plot

We observe in Figure 6 that the training loss decreases sharply while the validation loss remains relatively the same through the first three epochs. After that, the validation loss start rising significantly onwards. This is a clear sign of overfitting, which justifies our use of early stopping. It can also be explained because the model is already pre-trained well, and it only needs a few epochs to learn the new data. If we train the model over more epochs, it can easily overfit.

Figure 7 shows that during the fine-tuning phase, the validation accuracy steadily improves from 0.59 to 0.70 over 5 epochs. This indicates consistent gains in generalization as we fine-tune. However, we stop the training after the validation loss exceeds 0.9 as discussed. Therefore, the model is saved at the optimal point.

5.2 Test Set Performance

5.2.1 Classification Report

The model achieves an overall accuracy score of 0.7101, with an F1 score of the same (0.7101). The slightly higher scores for M (0.73 versus 0.69) reflect the slight class imbalance (347 M vs. 298 F), but overall the macro- and weighted-averages

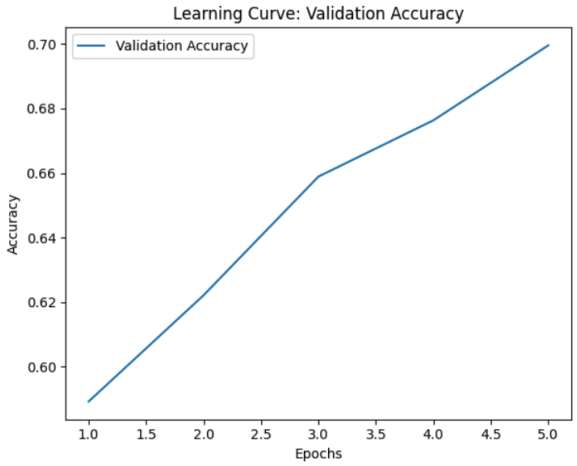

Figure 7: Fine-tuning Learning Curve

```
Evaluating on test set:
Loss: 1.0499
Accuracy: 0.7101
F1 Score: 0.7101
Classification Report:
              precision    recall  f1-score   support

           M       0.73      0.73      0.73       347
           F       0.69      0.69      0.69       298

    accuracy                           0.71       645
   macro avg       0.71      0.71      0.71       645
weighted avg       0.71      0.71      0.71       645
```

Figure 8: Classification Report

both settle at 0.71. These results confirm that our supervised contrastive pre-training plus fine-tuning pipeline delivers robust and equitable gender classification performance.

5.2.2 Confusion Matrix

Similar to the classification report, the confusion matrix (Figure 9) also confirms balanced performance across both classes.

6 Discussion

While the test performance does not significantly surpass traditional approaches like logistic regression or SVM in terms of raw accuracy or F1 score, our supervised contrastive learning pipeline demonstrates promising potential. The balanced performance across both classes and the model's ability to generalize after being trained on a relatively small dataset suggest that with further scaling or tuning, this approach could outperform classical methods. Additionally, the flexibility of the contrastive learning framework opens doors to incorporating richer semantic signals and more sophisticated augmentations in future work.

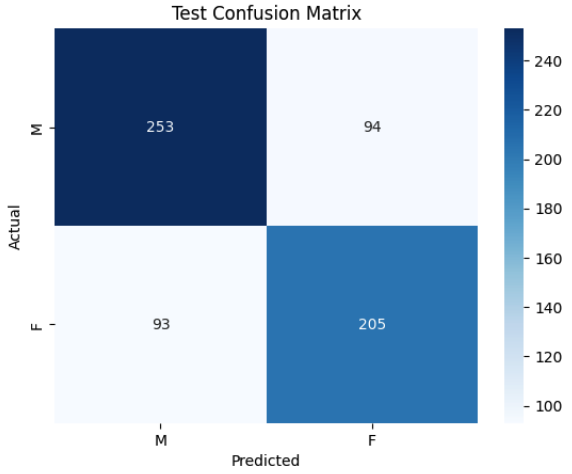

Figure 9: Confusion Matrix

7 Future Work

There are several avenues for future work. One is to experiment with different and more aggressive text augmentations for contrastive learning – for example, synonym replacement, shuffling sentence order, or back-translation – to see if they further improve the encoder's robustness. Another direction is to combine the contrastive loss and classification loss in a single joint training (multi-task learning) instead of sequentially. For instance, one could fine-tune the model on a weighted sum of contrastive loss and cross-entropy simultaneously; this might ensure that the learned features remain optimal for classification throughout training (some recent research suggests multi-task contrastive fine-tuning can boost performance).

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

J. Schler, M. Koppel, S. Argamon and J. Pennebaker. 2006. Blog authorship corpus.

https://www.kaggle.com/datasets/rtatman/blog-authorship-corpus.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Alexandre Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33.

Soujanya Mukherjee and Bing Liu. 2010. Predicting author gender from text. In *EMNLP*.

Tao Qian, Fei Li, Meishan Zhang, Guonian Jin, Ping Fan, and Wenhua Dai. 2022. Contrastive learning from label distribution: A case study on text classification. *Neurocomputing*, 507:208–220.

Daniel Wolf, Tristan Payer, Catharina Silvia Lisson, Christoph Gerhard Lisson, Meinrad Beer, Michael Götz, and Timo Ropinski. 2023. Self-supervised pre-training with contrastive and masked autoencoder methods for dealing with small datasets in deep learning for medical imaging. *Scientific Reports*, 13(1):20260.