

Learned Sparse Lexical Representations for Large-scale Authorship Attribution

Jian Zhu

Su Yuan

Changbing Yang

University of British Columbia

jian.zhu@ubc.ca, yuansu@student.ubc.ca,

cyang33@mail.ubc.ca

Abstract

Authorship attribution aims to determine whether a query text and a candidate document are written by the same author. We investigate high-dimensional learned sparse representations (LSR) and multi-vector retrieval for authorship attribution. We further propose Sparse ColBERT (SPARCOL), a late-interaction sparse lexical reranking model. To address the memory cost of vocabulary-size embeddings, we develop fused Triton kernels that can reduce peak memory usage by 5~11 times over the baseline torch implementation. Extensive experiments show that LSR can outperform dense embeddings in both single-genre and cross-genre settings, whereas SPARCOL improves reranking performance over single-vector models only in single-genre settings due to lexical matching bias. In addition to token-based embeddings, we also explore Lexicon-based embeddings that score clusters of tokens, which is competitive with the vocabulary-based LSR with reduced dimensions. We conduct interpretability analysis of LSRs, showing that they allow for a more transparent scoring process.

1 Introduction

The task of attribution of authorship is to determine whether texts are written by the same author. Low-dimensional dense embeddings have been shown to be effective for large-scale authorship attribution (Rivera-Soto et al., 2021; Zhu and Jurgens, 2021; Wegmann et al., 2022; Fincke and Boschee, 2024; Agarwal et al., 2025). Yet recently Weller et al. (2025) has demonstrated that dense embeddings are inherently limited by their low dimensions, and that high-dimensional sparse representations have strong potential in encoding richer text information. We hypothesize that using the high-dimensional vocabulary space might deliver extra performance than the low-dimensional dense embeddings.

Compared to dense embeddings, neural bag-of-words models like SPLADE (Formal et al., 2021)

are highly interpretable as its representations are token-based like the traditional bag-of-words approach that offers transparency on the scoring process. However, for authorship verification, especially in cross-genre scenarios, lexical overlaps are far less useful linguistic cues. So we ask: can a neural bag-of-words model like SPLADE or multi-vector retrieval model like ColBERT tackle cross-genre authorship verification?

In this study, we fill in the gap of high-dimensional learned sparse representations (LSR) and multi-vector retrieval in authorship attribution, which has rarely been applied to this use case. LSR builds sparse lexical embeddings that are highly transparent and accurate with neurally learned weights. We also propose Sparse ColBERT (SPARCOL), a late-interaction model that leverages token-level sparse top- k LSR for reranking. The architectures of the sparse single-vector and late-interaction models are shown in Figure 1.

LSR leads to the memory bottleneck due to the high-dimensional logit vectors. We develop custom Triton kernels to fuse the unembedding plus the pooling operations for SPLADE, and the unembedding plus top- k selection process for SPARCOL, reducing the memory usage by 5 ~ 11 times compared to PyTorch implementation by avoiding the full logits from materializing. These kernels make it possible to scale up the sequence length and batch size under the highly constrained academic computing resources.

Our findings show that LSR outperforms dense embeddings in both single-genre and cross-genre authorship attribution tasks. Multi-vector SPARCOL also outperforms single-vector retrieval models, especially in the single-genre settings, though the lexical matching biases significantly reduce cross-genre attribution performance. Since the tokenization vocabulary is often highly redundant, we also explore Lexicon-based embeddings that score on clusters of similar tokens, rather than individual

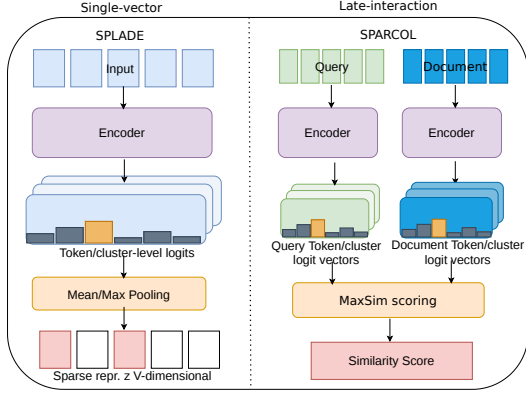


Figure 1: Comparison of sparse single-vector and late-interaction models. SPLADE first projects the input into token- or cluster-level logits and then pools them into a single sparse representation. In contrast, SPARCOL delays compression by retaining token-level sparse vectors for both the query and document texts and scoring them with MaxSim. Both token-based and cluster-based variants share the same overall architecture, differing only in the projection space.

tokens directly. Lexicon-based embeddings provide a good compromise between efficiency and the representation capacity brought by high dimensions.

Finally, interpretability analysis shows that while the token-level interpretation provided by these models is not deterministic, they do provide more transparency in comparing stylistic similarity.

For replication purposes, we will release our code and pretrained models at https://github.com/lingjzhu/neural_lexical.

2 Related works

2.1 Authorship verification

The use of dense embeddings in authorship analysis has been extensively investigated (Boenninghoff et al., 2019; Rivera-Soto et al., 2021; Zhu and Jurgens, 2021; Wegmann et al., 2022; Fincke and Boschee, 2024; Agarwal et al., 2025), which have demonstrated superior performance than traditional handcrafted features like character/word ngrams and TF-IDF vectors (Peng et al., 2003; Escalante et al., 2011; Sapkota et al., 2015).

Recent approaches also explore the use of LLMs in authorship attribution (Huang et al., 2024a,b, 2025; Tan et al., 2025). These approaches show promising results, but they are often limited to a few hundred authors, not scalable to large-scale scenarios where there are tens of thousands of can-

didate authors. Most current authorship verification research focuses on performance; few have investigated the interpretability of these neural methods. While LLMs can generate explanations for authorship attribution (Huang et al., 2024a; Ramnath et al., 2025), it is still unknown to what extent these explanations are faithful. In our approach, we attempt to train models that are inherently interpretable, addressing the faithful issue.

2.2 Sentence representations

Dense embeddings derived from sentence transformers (Reimers and Gurevych, 2019) are widely used in information retrieval because they are simple and effective. In parallel, learned sparse representations (LSRs), such as SPLADE and its variants (Formal et al., 2021; Lassance et al., 2024; Formal et al., 2024b; Doshi et al., 2024; Xu et al., 2025), project contextual hidden states into vocabulary dimensions and use the resulting bag-of-words-style vector as the representation. This design not only improves interpretability but also supports traditional large-scale retrieval infrastructure such as inverted indices. Building on this line of work, Lei et al. (2025) propose lexicon-based models that reduce the effective vocabulary size by clustering word embeddings and replacing individual vocabulary items with cluster centroids. Recent research also investigates the interpretability of sentence embeddings (Moeller et al., 2023, 2024; Opitz et al., 2026), but such works are still limited.

Another major line of research focuses on late-interaction models such as ColBERT (Khattab and Zaharia, 2020) and ColBERTv2 (Santhanam et al., 2022), which preserve token-level representations and perform fine-grained matching to improve retrieval quality. While many works have tried to bridge SPLADE and ColBERT to boost performance (Gao et al., 2021; Hai Le et al., 2023; Kong et al., 2023; Formal et al., 2024a), few works have attempted to directly use the vocabulary-sized embeddings for late interaction. Authorship representations are fundamentally different from the semantic representations in information retrieval. In information retrieval, most queries are short texts and candidates are long documents, whereas in authorship verification, the query and the candidates are symmetric with varying lengths, and lexical overlap might not be the most important cues. We address the gap in studying how LSR and late interaction can be applied to authorship analysis.

3 Method

We formulate authorship attribution as a retrieval problem, where a model needs to retrieve a text written by Author X from a pool of candidates, given a query written by the same Author X.

3.1 SPLADE

Given the final hidden states of the encoder, SPLADE produces sparse lexical representations through an unembedding operation followed by a non-linear activation and pooling. Let the final hidden states be denoted as $\mathbf{X} \in \mathbb{R}^{T \times D}$, and the tied embedding/unembedding matrix as $\mathbf{W} \in \mathbb{R}^{V \times D}$. The token-level vocabulary logits $\mathbf{Y} \in \mathbb{R}^{T \times V}$ are computed as

$$\mathbf{Y} = \mathbf{X}\mathbf{W}^\top \quad (1)$$

The logits will undergo a non-linear transformation:

$$\tilde{\mathbf{y}}_k = \log(1 + \sigma(\mathbf{y}_k)) \quad (2)$$

In the original SPLADE (Formal et al., 2021), σ is the ReLU activation function. However, we replaced the *ReLU* activation with ℓ_0 approximation activation (Shen et al., 2025):

$$\sigma(\mathbf{x}) = \log(1 + \max(0, \mathbf{x})) \quad (3)$$

Our consideration is more on the numerical side. Applying another round of log saturation can decrease the numerical range of logit values, reducing the gradient norm during training. The ReLU activation only makes the model not converge.

Then the final representation $\mathbf{z} = \text{Pool}(\tilde{\mathbf{Y}})$, where the pooling method could be max pooling or sum pooling. The model was trained contrastively with the InfoNCE loss (Bachman et al., 2019), or the multiple negative ranking loss (Henderson et al., 2017):

$$\mathcal{L}_I = -\mathbb{E} \left[\log \frac{\exp \left(\frac{\text{sim}(\mathbf{z}_i, \mathbf{z}_j)}{\tau} \right)}{\sum_{l=1}^N \exp \left(\frac{\text{sim}(\mathbf{z}_i, \mathbf{z}_l)}{\tau} \right)} \right] \quad (4)$$

where τ is the temperature and the $\text{sim}(\cdot)$ is a similarity function, typically cosine similarity for dense embeddings and dot product for sparse embeddings like SPLADE.

The original SPLADE (Formal et al., 2021; Lاسance et al., 2024) utilizes the FLOPS regularizer (Paria et al., 2020) to regularize the embedding sparsity. We set the λ to $5e - 3$.

$$\mathcal{L}_{flops} = \lambda \cdot \sum_{j \in V} \left(\frac{1}{N} \sum_{i=1}^N z_i \right)^2 \quad (5)$$

3.1.1 Fused kernels

Training SPLADE is far less efficient than conventional dense embeddings, due to the high dimensions of the vocabulary size. In the forward pass, the logits $\mathbf{Y}$ is a $T \times V$ matrix that will be immediately reduced to a fixed-dimensional vector $\mathbf{z} \in \mathbb{R}^{V \times 1}$ through pooling. In the mainstream pytorch implementation, $\mathbf{Y}$ needs to be cached in memory as a whole. This memory bottleneck is making it hard to scale up SPLADE, especially under the highly constrained academic compute.

The real question is: is the intermediate $\mathbf{Y}$ really necessary, since we only need the final lexical representation $\mathbf{h}$? To mitigate this bottleneck, we implemented fused kernels in Triton (Tillet et al., 2019) that fuse the unembedding and pooling operations, so that the logit vectors never materialize during forward and backward passes. The fused kernels take $\mathbf{X}$ and $\mathbf{W}$ as input and directly output the pooled vector.

3.1.2 Fused Max Pooling

In this kernel, $\mathbf{X}$ and $\mathbf{W}$ are partitioned into smaller tiles, and the local maximum is continuously updated as the kernel loops over matrix blocks. We also move all activations outside the kernel, as these operations are monotonic, such that they are unlikely to change the relative ranking between values.

3.1.3 Fused Sum Pooling

Similarly, the kernel keeps a running sum of the final output embeddings. Different from the Max Pooling kernel, ReLU activation is applied to each vector before summation to avoid negative logits. Note that this implementation differs slightly from the original SPLADE formulation. In the original SPLADE, the summation happens after the activations on each token’s logits. Here, we save the *log1p* operation outside the kernel for the sake of simplicity. As our queries can be up to 512 tokens long, we apply mean pooling to avoid gradient explosion due to the large logit values. For mean pooling, the final division of sequence length is done outside the kernel for simplicity.

Both kernels maintain the same precision as the pytorch implementation, while only costing a fraction of memory in both forward and backward

passes. We provide detailed derivation and benchmarking results in Appendix C.

3.2 SPARCOL

Late-interaction models exemplified by ColBERT (Khattab and Zaharia, 2020) and ColBERT V2 (Santhanam et al., 2022) can significantly improve the performance of information retrieval over single-vector encoders, at the cost of some computational and storage efficiency, though more efficient than cross-encoders.

Few studies have attempted to combine late interaction and LSR. Such a combination can make authorship verification interpretation at the token level and also offer fine-grained position-level interactions. Yet, combining these two seems pretty impractical, as it will be highly expensive to compute the high-dimensional logits for each token and more expensive to calculate the MaxSim scores. Here we only select the top- k logits per token $\mathbf{Y}_{top-k} \in \mathbb{R}^{T \times k}$ as representations, as prior works such as TopK SAE (Gao et al., 2025) suggest that it is possible to train high-dimensional sparse embeddings with top- k sparsity. To further improve training efficiency, we also developed two custom Triton kernels to work with top- k selection and scoring, with detailed descriptions and benchmarking in Appendix C.

3.2.1 Fused TopK

Rather than producing the full logit vectors, we developed a Triton kernel that fuses the unembedding and the top- k selection. We divide the hidden states $\mathbf{X}$ into chunks along sequence length T , and multiply with $\mathbf{W}$ by chunks, and only keep the top- k values and indices, such that the full logits will not fully materialize.

3.2.2 Sparse MaxSim

We compute the MaxSim on the top- k sparse representations. As Torch is not optimized for sparse vector multiplications, we remap the sparse query and document representations to truncated dense vectors whose dimensions are up to the number of unique tokens in the query batch. Since the number of unique tokens is always far less than the total vocabulary, MaxSim on these dense vectors will be more efficient. This allows us to leverage the highly optimized matmul operations while keeping the memory relatively low.

We did not apply the FLOPS loss to regularize the embedding, as that requires additional com-

putation on the full vocabulary embeddings. Instead, we simply enforced top- k sparsity with a fixed k_{train} . Since the pairwise comparison is done on the dot product as in SPLADE, we can select top- k embeddings with different k_{infer} values in inference time to achieve a trade-off between performance and efficiency.

3.3 Lexicon-based retrieval models

We adopt a Lexicon-based embedding approach similar to LENS (Lei et al., 2025), but with one key difference: instead of keeping cluster assignments fixed, we allow cluster membership to be updated during training.

In Lexicon-based embeddings, the final logits are computed with respect to the k cluster centroids $\mathbf{C} \in \mathbb{R}^{k \times D}$, which is initialized with k-means clustering on the embedding matrix.

$$\mathbf{Y}' = \mathbf{X}\mathbf{C}^\top \quad (6)$$

where $\mathbf{Y}' \in \mathbb{R}^{T \times k}$ is the actual logits that will be passed to the next step. During training, $\mathbf{C}$ is updated greedily. First, we recomputed the similarity of word embeddings with centroid embeddings.

$$\mathbf{S} = \mathbf{W}\mathbf{C}^\top. \quad (7)$$

where $\mathbf{S} \in \mathbb{R}^{V \times k}$ is the similarity matrix. Each embedding is then assigned to the cluster with the highest similarity

$$n_i = \arg \max_{j \in \{1, \dots, k\}} S_{i,j}. \quad (8)$$

and assign each token i to the cluster with the highest similarity. Then the centroid embedding $\mathbf{C}$ is recomputed using the new cluster assignments. Note that this approach does not guarantee that each cluster will be assigned at least one token. While running k-means, every few iterations can preserve k clusters, but it will incur too much computational cost. So we opted for the simple approach above.

We implemented the Lexicon-based variants of SPLADE-Lexicon with $k \in \{4000, 8000\}$ and SPARCOL-Lexicon with $k \in \{500, 1000, 2000\}$ and set the update frequency to every 10 steps. For SPLADE-Lexicon, we use the same Triton kernels to speed up training and inference. For SPARCOL-Lexicon, we only used dense matrix multiplications for MaxSim computation, which are faster with low-dimensional matrices, as the sparse kernel is only efficient when dimensions are high. To further reduce memory usage, we compute MaxSim incrementally in chunks.

4 Experiments

Datasets For the evaluation, we used the publicly available HRS1 and HRS2 English test sets from the HIATUS Program¹ hosted by IARPA. We reproduced the same test splits as those used in Agarwal et al. (2025) with random seeds {0, 1001, 2001, 3001}. The test set consists of two settings, CrossGenre and SingleGenre. In the SingleGenre condition, texts are all of the same genre (e.g., both are book reviews). In the more challenging CrossGenre condition, even texts from the same author can be from very different genres (recipe vs. forum posts).

For training, we used the same training dataset shared by Agarwal et al. (2025). The training dataset consists of over 200k pairs of the same-author texts. Texts written by the same authors were deliberately chosen to be different in topic (hard positives), with the semantic similarity smaller than 0.2 as given by all-MiniLM-L6-v2². Detailed statistics about training and evaluation datasets are available at Appendix A.

Training details We used ModernBERT (Warner et al., 2025) and Qwen3-0.6B (Yang et al., 2025) as the backbone models with tied embeddings. We also used Flash Attention 2 (Dao, 2024) and Liger Kernels (Hsu et al., 2024) (only for Qwen3) to optimize model training and inference. All SPLADE models were trained on an H100 20G MIG instance, and all ColBERT/SparCol models were trained on an H100 40G MIG instance. All were trained with a batch size of 64, except that SparCol was trained with 32 batch size, due to computational constraints. Detailed parameters are given in Appendix B.

Baselines We compared our models with several baselines from prior work (Agarwal et al., 2025) with the exact same evaluation setting. Agarwal et al. (2025) utilized complex hard negative mining strategies to boost performance. We did not use hard negative mining in our training. This also means our models are unlikely to outperform theirs, but their results can still provide references for the task. For fair comparison, we also include dense embeddings and ColBERT with the same base models and training data settings.

¹<https://www.iarpa.gov/research-programs/hiatus>

²<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

For ColBERT, we used the implementation from pylate (Chaffin and Sourty, 2025). We averaged the MaxSim score by the length of the query, which we found to stabilize training for long query texts; otherwise, the model failed to converge.

5 Results

We evaluate model performance with conventional retrieval metrics $Accuracy@k$, where $k \in \{8, 100\}$, and Mean reciprocal ranking at $k = 20$ ($MRR@20$), consistent with the official HIATUS metrics and other studies (Alshomary et al., 2025; Agarwal et al., 2025). For ColBERT and SPARCOL (-Lexicon), we evaluated them on reranking the top 100 candidates retrieved by the same backbone. Our main results are in Table 1 and Table 2, with detailed ablation analyses in Appendix D.

Comparison with BM25 and SPLADEv3 We compared with traditional token-based retrieval methods. BM25 and SPLADEv3 (Lassance et al., 2024), while originally only developed for semantic matching, can still distinguish authorship beyond chance level in the same genre (Refer to Table 2). But they are not able to attribute authors in cross-genre settings (See results in Table 1). This implies that token overlaps are important cues for authorship attribution in single-genre scenarios but not necessarily in cross-genre cases.

SPLADE outperforms dense embeddings in cross-genre authorship attribution but still remains competitive in single-genre settings Table 1 and 2 suggest that SPLADE remains competitive with dense embeddings, and outperforms dense embeddings by a large margin in cross-genre settings. Dense embeddings are still more efficient than SPLADE, as it saves the unembedding computation. Yet SPLADE does offer more transparency in the embedding.

Late-interaction reranker outperforms retrievers in single-genre condition but not in the cross-genre attribution Late-interaction rerankers have been understudied in authorship attribution. Our results confirm that multi-vector representations also work better than single-vector representations in authorship attribution, yet the advantage is more prominent in single-genre attributions.

SPARCOL outperforms ColBERT and SPLADE models at single-genre authorship attribution but does not have an advantage in cross-genre scenarios. SPARCOL relies on fine-grained token-level

Framework	Model / Params	Dim / Avg.NNZ	HRS1 / English Long			HRS2 / English Medium		
			Acc@8	Acc@100	MRR@20	Acc@8	Acc@100	MRR@20
Retriever								
BM25**	—	—	0.0	0.0	0.0	0.0	5.7	0.0
SBERT**	all-mpnet-base-v2 / 0.1B	768 / 768	0.2	3.1	0.3	0.7	3.4	0.2
SPLADEv3	splade-v3 / 0.1B	30522 / 325	0.0	1.3	0.0	0.0	4.0	0.0
SADIRI **	RoBERTa-large / 0.3B	1024 / 1024	26.0	62.8	14.8	34.0	58.2	18.6
SADIRI-v2 **	Mistral / 12B	2560 / 2560	42.1	84.9	24.0	59.9	84.2	45.6
Dense	ModernBERT-base / 0.1B	768 / 768	18.1	48.4	12.0	17.7	47.7	10.1
Dense	ModernBERT-large / 0.4B	1024 / 1024	14.5	42.1	8.4	20.5	58.2	11.5
SPLADE	ModernBERT-base / 0.1B	50368 / 1024	19.7	44.5	11.1	21.4	59.8	12.7
SPLADE	ModernBERT-large / 0.4B	50368 / 1024	26.4	59.4	14.8	37.0	70.6	20.9
SPLADE	Qwen3 / 0.6B	151936 / 1024	22.4	52.5	12.9	22.1	49.5	13.0
SPLADE-Lexicon	ModernBERT-large / 0.4B	4000 / 1024	25.1	64.3	13.5	33.6	65.2	20.0
SPLADE-Lexicon	ModernBERT-large / 0.4B	8000 / 1024	24.3	62.4	15.0	33.7	70.2	20.7
Crossencoder Reranker								
SADIRI-v2 Reranker**	Mistral 12B	—	48.3	-	31.1	68.4	-	53.1
Late-interaction Reranker								
ColBERT	ModernBERT / 0.1B	128 / 128	21.7	-	11.7	23.4	-	13.2
ColBERT	ModernBERT / 0.4B	128 / 128	23.6	-	14.9	32.9	-	22.5
SPARCOL	ModernBERT / 0.1B	50368 / 1024	14.9	-	9.7	21.4	-	13.6
SPARCOL	ModernBERT / 0.4B	50368 / 1024	22.8	-	12.3	28.2	-	20.9
SPARCOL-Lexicon	ModernBERT / 0.4B	500 / 102	22.3	-	13.0	27.5	-	15.4
SPARCOL-Lexicon	ModernBERT / 0.4B	1000 / 122	24.1	-	12.9	28.5	-	16.8
SPARCOL-Lexicon	ModernBERT / 0.4B	2000 / 188	22.5	-	14.3	27.5	-	17.7

Table 1: Cross-genre AA performance of retrieval-based models on the HRS1 and HRS2 datasets. NNZ stands for the number of non-zero entries. Ranker performance on the Top 100 candidates for each unique query. **Baseline results are from Agarwal et al. (2025). Note that SADIRI and SADIRI-v2 employed complex hard negative mining that we did not use, which accounts for the performance gap.

Framework	Model / Params	Dim / Avg.NNZ	HRS1 / English Long			HRS2 / English Medium		
			Acc@8	Acc@100	MRR@20	Acc@8	Acc@100	MRR@20
Retriever								
BM25	—	—	13.9	29.2	9.5	26.1	51.4	17.8
SBERT‡	all-mpnet-base-v2 / 0.1B	768 / 768	6.5	24.9	3.5	19.8	52.5	9.9
SPLADEv3	splade-v3 / 0.1B	30522 / 325	11.9	27.9	6.4	19.0	46.9	10.7
SADIRI-v2 ‡	Mistral / 12B	2560 / 2560	74.7	95.1	53.6	88.3	97.7	72.8
Dense	ModernBERT-base / 0.1B	768 / 768	51.7	74.5	33.8	64.6	89.6	46.4
Dense	ModernBERT-large / 0.4B	1024 / 1024	54.4	76.1	37.9	69.2	90.7	49.9
SPLADE	ModernBERT-base / 0.1B	50368 / 1024	47.7	73.2	30.7	63.8	89.6	45.1
SPLADE	ModernBERT-large / 0.4B	50368 / 1024	47.9	76.9	31.4	69.1	93.0	48.9
SPLADE	Qwen3 / 0.6B	151936 / 1024	54.2	80.2	35.9	66.6	87.8	46.4
SPLADE-Lexicon	ModernBERT-large / 0.4B	4000 / 1024	62.6	86.4	43.3	79.7	95.7	60.4
SPLADE-Lexicon	ModernBERT-large / 0.4B	8000 / 1024	58.3	81.9	38.4	77.4	94.9	56.3
Late-interaction Reranker								
ColBERT	ModernBERT-base / 0.1B	128 / 128	57.8	-	40.3	70.4	-	51.5
ColBERT	ModernBERT-base / 0.4B	128 / 128	60.8	-	42.5	71.5	-	52.6
SPARCOL	ModernBERT-base / 0.1B	50368 / 1024	58.0	-	40.5	74.5	-	56.9
SPARCOL	ModernBERT-base / 0.4B	50368 / 1024	66.5	-	48.8	81.4	-	64.6
SPARCOL-Lexicon	ModernBERT / 0.4B	500 / 102	59.7	-	43.6	74.8	-	55.6
SPARCOL-Lexicon	ModernBERT / 0.4B	1000 / 122	61.2	-	44.6	74.9	-	58.0
SPARCOL-Lexicon	ModernBERT / 0.4B	2000 / 188	63.2	-	47.6	78.1	-	61.7

Table 2: Single-genre AA performance of retrieval-based models on the HRS1 and HRS2 datasets. NNZ stands for the number of non-zero entries. Ranker performance on Top 100 candidates for each unique query. ‡ We reported these numbers by running these models as shared by Agarwal et al. (2025).

matching with bag-of-words vectors. It works well for single-genre attribution where token matching

plays an important role, as the MaxSim scoring in SparCol amplifies the stylistic matching signals.

Dim	Initialization	After training
2687	"sid"	"_subsid", "_Syd", "_Sid", "_sid", "sid", "_Sidney", "SID"
2710	"_bool", "BOOL"	"_BOOL", "_uint", "_bool", "uint", "_uintptr", "bool", "_boolean", "boolean", "_Boolean", "Boolean", "Bool", "BOOL", "UInt", "UInt"
2719	"umer"	"um", "ume", "uman", "UM", "ums", "umer", "umes", "uma", "uming", "ummy", "oler", "umm", "umed", "umar", "umen"
1537	"_Krish"	"_Raj", "_Delhi", "_Singh", "_Chand", "_Mumbai", "_Modi", "_Kumar", "_Pradesh", "_BJP", "_Gandhi", "rish", "_Krish", "_Karn", "_Punjab"

Table 3: Token clusters initialized from k-means initialization evolved into more meaningful clusters, as tokens sharing character ngrams, cultural reference, and meanings are grouped together.

Yet its inductive bias does not suit the cross-genre scenarios where token matching no longer works and higher-level stylistic cues (beyond the lexical level) must be mined. Dense ColBERT performs token matching in the latent space, which allows for the encoding of abstract cues.

Even with reduced dimensions, Lexicon-based models can match the performance of token-based embeddings Lexicon-based embeddings can match the performance of token-based embeddings closely, if not exceeding them, in both retriever and late-interaction rerankers, suggesting that the vocabulary dimensions are highly redundant. Meanwhile, with reduced dimensions, Lexicon-based embeddings are more efficient to train and inference, which can be promising for future attribution systems. Yet, if the Lexicon dimensions are too low, like in SPARCOL-Lexicon, the performance is still lower than the high-dimensional token embeddings. In Table 3, it can be seen that tokens are merged with other tokens of a similar nature, such as shared character ngram, similar token variants, and similar cultural references.

6 Analyzing sparse representations

6.1 SPLADE

SPLADE is argued to be more interpretable than dense embeddings, as it produces a bag-of-words vector with learned weights. The interpretability is offered by tied embeddings, where the unembedding matrix and the input embedding matrices are the same. SPLADE will automatically perform query expansion, assigning weights to tokens not in the original texts (Mackenzie et al., 2023), in contrast to TF-IDF vectors in BM25 that limits to tokens in text. Yet authorship SPLADE relies more on query expansion than the semantic SPLADE-v3, as input tokens only account for around 7% of total activation mass (vs. 30% in SPLADE-v3).

Style probing To understand how styles are represented in lexical models, we performed style probing using SynthSTEL (Patel et al., 2025), a synthetic stylistic benchmark covering 40 stylistic dimensions. In each SynthSTEL pair, the two texts only differ in one stylistic dimension but share the same semantic content. We encoded each SynthSTEL pair with SPLADE variants. Binary logistic regression probes were trained on the top 1024 active final SPLADE dimensions per example, using pair-level 80/20 train/test random splits. Results are averaged across 10 runs.

Table 5 shows that SPLADE fine-tuned for authorship attribution can reliably distinguish styles with a higher success rate than SPLADE-v3 (Lassance et al., 2024). In Table 4, SPLADE models can represent distinct styles with interpretable lexical features, especially in capturing surface forms and function words, though at times interpretations might not be straightforward. For example, SPLADE-v3 performs at random for capitalization features, but all authorship attributions achieve perfect performance with correct lexical features.

Despite the bag-of-words constraint that limits embeddings to concrete tokens or token clusters, SPLADE(-Lexicon) embeddings can still store abstract features through enumerating tokens of a similar nature in the vocabulary. For example, Arabic numbers in the original text will also lead to high activation values in token clusters of Arabic numbers. However, these interpretations are not deterministic, as even with the same training data, SPLADE and SPLADE-Lexicon will give different weights to different types of tokens, suggesting that such interpretations can still be limited. Examples for both SPLADE and SPLADE-Lexicon are available in Appendix F.

6.2 SparCol

SPARCOL (-Lexicon) emphasizes local token matching, as a query token, and only matches to

Style Feature	SPLADE-v3 (semantic)	SPLADE-Base	SPLADE-Large	SPLADE-Lexicon-Large
Self / Third-Person Singular	68.7 / 82.0 his he she her no	87.5 / 93.3 he I she me hers	79.3 / 85.0 He He his she Your	76.0 / 83.5 c_1023: he he his her we she He him We We
With / Without Sarcasm	62.8 / 68.1 no [SEP] oh joy wow	79.2 / 87.9 , iber { } , ! faculty	82.5 / 90.6 ol hyp atom ap films	80.5 / 86.7 c_388: ight exact perfect exactly straight perfectly precisely precise Exactly
With / Without All Upper Case	50.0 / 50.0 [PAD] strolled spines biotechnology sharif	100.0 / 100.0 ITY FLAG FUN TRAN LET	100.0 / 100.0 SU THE FOR ONE OO	100.0 / 100.0 c_1111: LE IL NE BE NE LE

Table 4: SynthSTEL linear-probing results by styles. Each cell reports accuracy / AUC / top lexical features. A full list of features is available in Appendix E.

Model	Accuracy	AUC
SPLADE-v3 (semantic)	64.4	70.7
SPLADE-Base	86.6	92.9
SPLADE-Large	84.4	90.6
SPLADE-Lexicon-Large	82.1	88.8

Table 5: SynthSTEL linear probing results.

Model	Top-1 Accuracy	AUC
Adversarial		
SPLADE-v3 (semantic)	3.1	50.8
SPLADE-Base	25.1	60.7
SPLADE-Large	17.2	55.8
SPLADE-Lexicon-Large	23.5	56.7
Non-adversarial		
SPLADE-v3 (semantic)	54.5	51.4
SPLADE-Base	74.5	61.3
SPLADE-Large	69.8	56.4
SPLADE-Lexicon-Large	62.7	57.2
ColBERTv2 (semantic)	69.60	56.11
SPARCOL-large	62.33	52.84
SPARCOL-large-Lexicon	68.46	56.77

Table 6: Style retrieval performance on SynthSTEL using both positive and negative query classes. In adversarial conditions, sentences with the same meaning but different styles are included.

a document token through MaxSim scoring. Due to the multi-vector representation, linear probing does not easily apply. We reformulated SynthSTEL (Patel et al., 2025) as a retrieval task where each model had to retrieve the sentences with the same style correctly. In the adversarial condition, we included the same-meaning-different-style negatives as a test to test if the models can disentangle meaning and style. Table 6 shows that all models exhibit a large performance drop in the adversarial condition. We additionally include ColBERTv2 for semantic retrieval (Santhanam et al., 2022).

Yet SPLADEs tend to outperform SPARCOL

models across conditions, suggesting that single-vector models achieve stronger disentanglement of content and style. Such findings also match the results for single-genre and cross-genre attribution, where SPARCOL performs worse at cross-genre attributions. The lexical matching biases in multi-vector retrieval models make them hard to separate content from style. On the other hand, while it is common knowledge that separation of content and style is critical for authorship tasks (Wegmann et al., 2022; Patel et al., 2025), our finding implies that a perfect separation is not always needed, especially in the single-genre attribution.

We include an example in Table 27, 28, 29 and 30 in the Appendix. In those examples, exact matching happens more frequently with function words and punctuation as they are highly frequent in all texts. This inductive bias also explains the performance gap of SparCol between cross-genre and single-genre scenarios. SparCol is particularly strong when there are exact matches of tokens between texts, but still struggles to encode abstract stylistic features that transcend simple lexical matching.

7 Conclusion

In this study, we investigate the role of learned sparse representations and Lexicon-based embeddings in authorship attributions. We developed custom Triton kernels to improve the memory efficiency of the original SPLADE and further propose SparCol, a late-interaction learned sparse representation model. Our findings demonstrate that learned sparse representations can improve the performance of authorship attributions while providing more transparent embeddings. Together, these findings suggest that sparsity and late interactions provide a promising direction for more interpretable authorship attributions.

8 Limitations

Learned sparse representations, while highly interpretable, are still not as efficient as common dense embeddings, at least in the current paradigm. Even though we have made non-trivial optimizations on the computing efficiency, we believe that there is still a large room for improvement. Our paper centers on optimizing the training efficiency rather than inference efficiency. For sparse models like SPLADE, traditional methods like inverted indices can still be utilized for large-scale deployment. For SPARCOL, techniques like quantization and pruning can further improve the inference efficiency.

While our research provides more transparent embeddings than dense embeddings. The sparse embeddings are still far from being directly human interpretable, as many activations are localized to specific words or specific locations. Many activated tokens or token clusters also do not have direct associations with the given text. Hence, the interpretability is still limited. We will continue to investigate how to convert these to better interpretations that can aid human decision-making.

9 Ethical considerations

Our research focuses on the algorithmic improvement of authorship attribution. By exploring how authorship attribution can be made more interpretable, our research contribute to the transparency and trustworthiness of the attribution decision making process. While authorship attribution can be applied in many critical areas such as forensic linguistics and social media bot identification, malicious use of this technology can potentially compromise user privacy. We urge ethical deployment of such technology.

LLM Usage We acknowledge the use of LLMs to assist with grammar correction and polishing based on a written draft. LLMs were also used for general coding assistance.

Acknowledgments

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract 202222072200006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of

ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

This research was enabled in part through the computational resources provided by Advanced Research Computing at the University of British Columbia and the Digital Research Alliance of Canada. The research activities were also supported by the NSERC Discovery Grant and the CFI-JELF Grant awarded to JZ.

References

- Shantanu Agarwal, Joel Barry, Steven Fincke, and Scott Miller. 2025. Cross-genre authorship attribution via llm-based retrieve-and-rerank. *arXiv preprint arXiv:2510.16819*.
- Milad Alshomary, Nikhil Reddy Varimalla, Vishal Anand, Smaranda Muresan, and Kathleen McKeown. 2025. [Layered insights: Generalizable analysis of human authorial style by leveraging all transformer layers](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10279–10292, Suzhou, China. Association for Computational Linguistics.
- Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32.
- Benedikt Boenninghoff, Robert M Nickel, Steffen Zeiler, and Dorothea Kolossa. 2019. Similarity learning for authorship verification in social media. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2457–2461. IEEE.
- Antoine Chaffin and Raphaël Sourty. 2025. Pylate: Flexible training and retrieval for late interaction models. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 6334–6339.
- Tri Dao. 2024. [Flashattention-2: Faster attention with better parallelism and work partitioning](#). In *The Twelfth International Conference on Learning Representations*.
- Meet Doshi, Vishwajeet Kumar, Rudra Murthy, Jaydeep Sen, and 1 others. 2024. Mistral-splade: Llm for better learned sparse retrieval. *arXiv preprint arXiv:2408.11119*.
- Hugo Jair Escalante, Tamar Solorio, and Manuel Montes-y Gómez. 2011. [Local histograms of character n-grams for authorship attribution](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 288–298, Portland, Oregon, USA. Association for Computational Linguistics.

- Steven Fincke and Elizabeth Boschee. 2024. Separating style from substance: Enhancing cross-genre authorship attribution through data selection and presentation. *arXiv preprint arXiv:2408.05192*.
- Thibault Formal, Stéphane Clinchant, Hervé Déjean, and Carlos Lassance. 2024a. Splade: Sparse late interaction retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2635–2640.
- Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.
- Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2024b. Towards effective and efficient sparse neural information retrieval. *ACM Transactions on Information Systems*, 42(5):1–46.
- Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2025. [Scaling and evaluating sparse autoencoders](#). In *The Thirteenth International Conference on Learning Representations*.
- Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. [COIL: Revisit exact lexical match in information retrieval with contextualized inverted list](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3030–3042, Online. Association for Computational Linguistics.
- Nam Hai Le, Thomas Gerald, Thibault Formal, Jian-Yun Nie, Benjamin Piwowarski, and Laure Soulier. 2023. Cosplade: Contextualizing splade for conversational information retrieval. In *European Conference on Information Retrieval*, pages 537–552. Springer.
- Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.
- Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, and Yanning Chen. 2024. Liger kernel: Efficient triton kernels for llm training. *arXiv preprint arXiv:2410.10989*.
- Baixiang Huang, Canyu Chen, and Kai Shu. 2024a. [Can large language models identify authorship?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 445–460, Miami, Florida, USA. Association for Computational Linguistics.
- Baixiang Huang, Canyu Chen, and Kai Shu. 2025. Authorship attribution in the era of llms: Problems, methodologies, and challenges. *ACM SIGKDD Explorations Newsletter*, 26(2):21–43.
- Weihang Huang, Akira Murakami, and Jack Grieve. 2024b. Alms: Authorial language models for authorship attribution. *arXiv preprint arXiv:2401.12005*.
- Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.
- Weize Kong, Jeffrey M Dudek, Cheng Li, Mingyang Zhang, and Michael Bendersky. 2023. Sparseembed: Learning sparse lexical representations with contextual embeddings for retrieval. In *Proceedings of the 46th International ACM SIGIR conference on research and development in information retrieval*, pages 2399–2403.
- Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. Splade-v3: New baselines for splade. *arXiv preprint arXiv:2403.06789*.
- Yibin Lei, Tao Shen, Yu Cao, and Andrew Yates. 2025. [Enhancing lexicon-based text embeddings with large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18986–19001, Vienna, Austria. Association for Computational Linguistics.
- Joel Mackenzie, Shengyao Zhuang, and Guido Zuccon. 2023. Exploring the representation power of splade models. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 143–147.
- Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *arXiv preprint arXiv:2509.06888*.
- Lucas Moeller, Dmitry Nikolaev, and Sebastian Padó. 2023. [An attribution method for Siamese encoders](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15818–15827, Singapore. Association for Computational Linguistics.
- Lucas Moeller, Dmitry Nikolaev, and Sebastian Padó. 2024. [Approximate attributions for off-the-shelf Siamese transformers](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2059–2071, St. Julian’s, Malta. Association for Computational Linguistics.
- Juri Opitz, Andrianos Michail, Lucas Moeller, Sebastian Padó, and Simon Clematide. 2026. [Similar, but why? a toolkit for explaining text similarity](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 203–214, Rabat, Morocco. Association for Computational Linguistics.

- Biswajit Paria, Chih-Kuan Yeh, Ian E.H. Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. 2020. [Minimizing flops to learn efficient sparse representations](#). In *International Conference on Learning Representations*.
- Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Chris Callison-Burch. 2025. [StyleDistance: Stronger content-independent style embeddings with synthetic parallel examples](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8662–8685, Albuquerque, New Mexico. Association for Computational Linguistics.
- Fuchun Peng, Dale Schuurmans, Vlado Keselj, and Shaojun Wang. 2003. [Language independent authorship attribution with character level n-grams](#). In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. Association for Computational Linguistics.
- Sahana Ramnath, Kartik Pandey, Elizabeth Boschee, and Xiang Ren. 2025. [CAVE: Controllable authorship verification explanations](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8939–8961, Albuquerque, New Mexico. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. [Learning universal authorship representations](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. [ColBERTv2: Effective and efficient retrieval via lightweight late interaction](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.
- Upendra Sapkota, Steven Bethard, Manuel Montes, and Tamar Solorio. 2015. [Not all character n-grams are created equal: A study in authorship attribution](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–102, Denver, Colorado. Association for Computational Linguistics.
- Xinjie Shen, Zhichao Geng, and Yang Yang. 2025. [Exploring 0 sparsification for inference-free sparse retrievers](#). In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '25*, page 2572–2576, New York, NY, USA. Association for Computing Machinery.
- Xinhao Tan, Songhua Liu, Xia Cong, Kunjun Li, and Xinchao Wang. 2025. [Open-world authorship attribution](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17744–17758, Vienna, Austria. Association for Computational Linguistics.
- Philippe Tillet, Hsiang-Tsung Kung, and David Cox. 2019. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 10–19.
- Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.
- Anna Wegmann, Dong Nguyen, and David Jurgens. 2025. [Tokenization is sensitive to language variation](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10958–10983, Vienna, Austria. Association for Computational Linguistics.
- Anna Wegmann, Marijn Schraagen, and Dong Nguyen. 2022. [Same author or just same topic? towards content-independent style representations](#). In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 249–268, Dublin, Ireland. Association for Computational Linguistics.
- Orion Weller, Michael Boratko, Iftexhar Naim, and Jinhyuk Lee. 2025. On the theoretical limitations of embedding-based retrieval. *arXiv preprint arXiv:2508.21038*.
- Zhichao Xu, Aosong Feng, Yijun Tian, Haibo Ding, and Lin Lee Cheong. 2025. Csplade: Learned sparse retrieval with causal language models. *arXiv preprint arXiv:2504.10816*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jian Zhu and David Jurgens. 2021. *Idiosyncratic but not arbitrary: Learning idiolects in online registers reveals distinctive yet consistent individual styles*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 279–297, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Dataset

A.1 Training Data

Table 7 summarizes the training data sources used in our experiments. The data was shared by Agarwal et al. (2025) and consists of diverse text genres including news articles, books, social media posts, and academic publications, all from public sources and anonymized properly.

Source	Type	Docs	Authors
RealNews	News	191,020	95,510
BookCorpus	Books	66,786	33,393
Reddit	Posts	55,014	27,507
GoodReads	Reviews	34,010	17,005
Amazon	Reviews	17,476	8,738
Gmane	Newsgroups	16,396	8,198
WikiDisc	Talk pages	13,724	6,862
PubMed	Articles	692	346
Total		395,118	197,559

Table 7: Training data sources with document and author counts.

A.2 Evaluation Data

We evaluate on the HIATUS benchmark, which includes two test sets (HRS1 and HRS2) spanning multiple genres. Table 8 provides an overview of both test sets, and Table 9 details the genre-level distribution.

Test Set	Foreground		Background	
	Docs	Authors	Docs	Authors
HRS1	855	114	37,084	16,313
HRS2	631	37	10,842	2,581
Total	1,486	151	47,926	18,894

Table 8: Overview of HIATUS evaluation sets.

B Training details

Details about training hyperparameters are available in Table 16, 12, 13, 14, and 15.

C Benchmarking Triton kernels

Here we provide the detailed derivation of the forward and the backward passes, which are then implemented in Triton kernels. All kernels outputs and

Set	Genre	FG	BG
HRS1	Board game reviews	212	2,432
	Citizens journalism	140	5,453
	Instructions	133	8,534
	Literature forums	199	9,641
	STEM forums	171	11,024
HRS2	Posts on pets	112	1,298
	Personal tales	138	1,542
	Product reviews	121	0
	Obituaries	132	3,625
	Movie reviews	128	4,377

Table 9: Genre distribution across HIATUS test sets. FG = Foreground docs, BG = Background docs.

Hyperparameters	Value
Base Encoder	Qwen3-0.6B
Embedding Dimension	1024
Vocabulary Size	151,936
Pooling Strategy	Mean pooling
Optimizer	AdamW
Learning Rate	5×10^{-5}
Weight Decay	1×10^{-4}
LR Scheduler	Cosine
Warmup Steps	0
Batch Size	64
Max Sequence Length	512
Mixed Precision	bfloat16
Training Steps	12,652
Hardware	H100 MIG 20GB
Time	15h 40m
Memory	19GB

Table 10: SPLADE Qwen3 training configuration.

gradients achieve cosine similarities ≥ 0.99 with the original PyTorch implementation, implying that they align well with torch implementation, so we only provided the latency and memory benchmarking.

C.1 Fused Max Pooling

In this kernel, $\mathbf{X}$ and $\mathbf{W}$ are partitioned into smaller tiles, and the local maximum is continuously updated as the kernel loops over matrix blocks. We also move the $\log(1 + \sigma(\cdot))$ outside the kernel, as these operations are monotonic such that they are unlikely to change the relative ranking between values.

For each batch element $b \in \{1, \dots, B\}$, token position $t \in \{1, \dots, T\}$, and vocabulary index $v \in \{1, \dots, V\}$, we compute unembedding scores

$$s_{b,t,v} = X_{b,t}^\top W_v, \quad (9)$$

where $W_v \in \mathbb{R}^D$ denotes the v -th row of the unembedding matrix.

We then apply max-pooling over the token di-

Hyperparameters	Value
Base Encoder	ModernBERT large
Cluster Size	8000 / 4000
Pooling Strategy	Mean pooling
Optimizer	AdamW
Learning Rate	5×10^{-5}
Weight Decay	1×10^{-4}
LR Scheduler	Cosine
Warmup Steps	0
Batch Size	128
Max Sequence Length	512
Mixed Precision	bfloat16
Training Steps	3120
Hardware	H100 MIG 40GB
Time	7h 20m / 6h 40m
Memory	25 GB

Table 11: SPLADE-Lexicon training configuration.

Hyperparameters	Value
Base Encoder	ModernBERT-base / large
Embedding Dimension	768 / 1024
Vocabulary Size	50,368
Pooling Strategy	Mean pooling
Optimizer	AdamW
Learning Rate	$1 \times 10^{-4} / 5 \times 10^{-5}$
Weight Decay	1×10^{-4}
LR Scheduler	Cosine
Warmup Steps	0
Batch Size	64
Max Sequence Length	512
Mixed Precision	bfloat16
Training Steps	6,326
Hardware	H100 MIG 20GB
Time	3h 13m / 5h 56m
Memory	11GB / 19GB

Table 12: SPLADE ModernBERT training configuration.

mension:

$$y_{b,v} = \max_{t \in \{1, \dots, T\}} s_{b,t,v}. \quad (10)$$

We additionally record the maximizing token index

$$t_{b,v}^* = \arg \max_{t \in \{1, \dots, T\}} s_{b,t,v}. \quad (11)$$

The forward operator thus produces pooled vocabulary scores $y \in \mathbb{R}^{B \times V}$ and argmax indices $t^* \in \mathbb{N}^{B \times V}$.

Backward Pass. Let $g_{b,v} = \frac{\partial \mathcal{L}}{\partial y_{b,v}}$ denote the upstream gradient from the loss $\mathcal{L}$. The max-pooling operation produces sparse gradients that propagate only through the tokens with maximum values.

Gradient w.r.t. Token Representations. For each batch index b and token position t ,

$$\frac{\partial \mathcal{L}}{\partial X_{b,t}} = \sum_{v=1}^V \mathbb{I}[t = t_{b,v}^*] g_{b,v} W_v, \quad (12)$$

Hyperparameters	Value
Base Encoder	ModernBERT-base / large
Embedding Dimension	768 / 1024
Vocabulary Size	50,368
Pooling Strategy	Mean pooling
Optimizer	AdamW
Learning Rate	3×10^{-4}
Weight Decay	0
LR Scheduler	Cosine
Warmup Steps	0
Batch Size	64
Max Sequence Length	512
Mixed Precision	bfloat16
Training Steps	6,326
Hardware	H100 MIG 20GB
Time	1h 40m / 3h 52m
Memory	10GB / 18GB

Table 13: Dense ModernBERT training configuration.

Hyperparameters	Value
Base Encoder	ModernBERT-base / large
Embedding Dimension	768 / 1024
Vocabulary Size	50,368
Similarity Function	MaxSim
Optimizer	AdamW
Learning Rate	$1 \times 10^{-4} / 5 \times 10^{-5}$
Weight Decay	0
LR Scheduler	Cosine
Warmup Steps	1,000
Batch Size	64
Max Sequence Length	512
Mixed Precision	fp16
Training Epochs	2
Hardware	RTX 5000 Ada (32GB)
Time	2h 1m / 3h 55m
Memory	26GB /

Table 14: ColBERT ModernBERT training configuration.

where $\mathbb{I}[\cdot]$ is the indicator function.

Gradient w.r.t. Unembedding Matrix. For each vocabulary index v ,

$$\frac{\partial \mathcal{L}}{\partial W_v} = \sum_{b=1}^B g_{b,v} X_{b,t_{b,v}^*}. \quad (13)$$

C.2 Fused Sum Pooling

Different from the Max Pooling kernel, ReLU activation is applied to each vector before summation. Similarly, the kernel keeps a running sum of the final output embeddings. Note that this implementation differs from the original SPLADE formulation. In the original SPLADE, the summation happens after the $\log(1 + \sigma(\cdot))$ operations on each token logits. Here we save the $\log 1p$ operation outside the kernel for the sake of simplicity.

Forward Pass. For each batch element $b \in \{1, \dots, B\}$, token index $t \in \{1, \dots, T\}$, and vo-

Hyperparameters	Value
Base Encoder	ModernBERT-base / large
Embedding Dimension	768 / 1024
Vocabulary Size	50,368
Top-k	1024
Optimizer	AdamW
Learning Rate	5×10^{-5}
Weight Decay	$5 \times 10^{-5} / 1 \times 10^{-2}$
LR Scheduler	Cosine
Warmup Ratio	0.15
Batch Size	64 / 64
Max Sequence Length	512
Mixed Precision	fp16
Training Epochs	2
Hardware	H100 MIG 40G
Time	7h 10m / 12h 8m
Memory	31GB / 38GB

Table 15: SPARCOL training configuration.

Hyperparameters	Value
Base Encoder	ModernBERT large
Cluster Size	500 / 1000 / 4000
Optimizer	AdamW
Learning Rate	5×10^{-5}
Weight Decay	1×10^{-4}
LR Scheduler	Cosine
Warmup Steps	0
Batch Size	64
Max Sequence Length	512
Mixed Precision	bfloat16
Training Steps	3120
Hardware	H100 MIG 40GB
Time	7h / 9h / 14h
Memory	31 GB

Table 16: SPARCOL-Lexicon training configuration.

cabulary index $v \in \{1, \dots, V\}$, we compute unembedding logits with ReLU activation.

$$z_{b,t,v} = \max(0, X_{b,t}^\top W_v) \quad (14)$$

The output features for each vocabulary entry is obtained by summing over the token dimension:

$$y_{b,v} = \sum_{t=1}^T z_{b,t,v}. \quad (15)$$

Also, we save a binary ReLU activation mask

$$m_{b,t,v} = \mathbb{I}[z_{b,t,v} > 0], \quad (16)$$

because gradients will be 0 for non-positive entries after ReLU.

Backward Pass The backward pass is also straightforward as the forward pass only involves a matrix multiplication and a ReLU function.

Gradient w.r.t. last hidden states The gradient with respect to token embeddings is given by

$$\frac{\partial \mathcal{L}}{\partial X_{b,t}} = \sum_{v=1}^V g_{b,v} m_{b,t,v} W_v. \quad (17)$$

Only vocabulary entries whose unembedding activation was positive contribute to the gradient.

Gradient w.r.t. unembedding matrix. The gradient with respect to the unembedding matrix is

$$\frac{\partial \mathcal{L}}{\partial W_v} = \sum_{b=1}^B \sum_{t=1}^T g_{b,v} m_{b,t,v} X_{b,t}. \quad (18)$$

C.3 Fused TopK

Rather than producing the full logit vectors, we developed a Triton kernel that fuses the unembedding and the TopK logit entries selection, so that the full logits never materialize.

Forward Pass. For each batch index $b \in \{1, \dots, B\}$, token index $t \in \{1, \dots, T\}$, and vocabulary index $v \in \{1, \dots, V\}$, we compute unembedding scores

$$z_{b,t,v} = X_{b,t}^\top W_v. \quad (19)$$

For each token (b, t) , we select the k largest unembedding scores over the vocabulary dimension:

$$\mathcal{K}_{b,t} = \text{TopK}_k \left(\{z_{b,t,v}\}_{v=1}^V \right), \quad (20)$$

where $\mathcal{K}_{b,t} \subset \{1, \dots, V\}$ denotes the set of selected vocabulary indices.

We denote the selected values and indices as

$$y_{b,t,i} = z_{b,t,v_i}, \quad (21)$$

$$v_i \in \mathcal{K}_{b,t}, \quad i = 1, \dots, k. \quad (22)$$

The forward operator thus outputs $y \in \mathbb{R}^{B \times T \times k}$ and the corresponding vocabulary indices $\mathcal{K} \in \mathbb{N}^{B \times T \times k}$.

Backward Pass. Let $g_{b,t,i} = \frac{\partial \mathcal{L}}{\partial y_{b,t,i}}$ denote the upstream gradient. In the Top- k operation, we only have gradients in the selected vocabulary entries.

Gradient w.r.t. Token Representations. For each token (b, t) , the gradient with respect to its representation is

$$\frac{\partial \mathcal{L}}{\partial X_{b,t}} = \sum_{i=1}^k g_{b,t,i} W_{v_i}, \quad v_i \in \mathcal{K}_{b,t}. \quad (23)$$

Gradient w.r.t. Unembedding Matrix. For each vocabulary index v , the gradient is given by

$$\frac{\partial \mathcal{L}}{\partial W_v} = \sum_{b=1}^B \sum_{t=1}^T \sum_{i=1}^k \mathbb{I}[v = v_i] g_{b,t,i} X_{b,t}, \quad (24)$$

where $\mathbb{I}[\cdot]$ is the indicator function, which will be 1 for the selected entries and 0 otherwise.

C.4 Sparse MaxSim

We compute the MaxSim on the sparse TopK sparse representations. As Torch is not highly optimized for sparse vector multiplications, we project the sparse representations to truncated dense vectors that only contain the unique tokens in the query batch. This allows us to leverage the highly optimized matmul operations while keeping the memory relatively low.

Let

$$Q^{(b_1)} = \{(q_{b_1,t,k}, i_{b_1,t,k}^Q)\}_{k=1}^K \quad \text{for } t = 1, \dots, T_q$$

denote the sparse query representation for query batch index b_1 , and

$$D^{(b_2)} = \{(d_{b_2,s,k}, i_{b_2,s,k}^D)\}_{k=1}^K \quad \text{for } s = 1, \dots, T_d$$

denote the sparse document representation for document batch index b_2 .

Each token is represented by K nonzero values with associated vocabulary indices.

Sparse Token Similarity. The similarity between query token t and document token s is defined as

$$\text{sim}(t, s) = \sum_{k=1}^K \sum_{\ell=1}^K \mathbb{I}[i_{b_1,t,k}^Q = i_{b_2,s,\ell}^D] q_{b_1,t,k} d_{b_2,s,\ell}. \quad (25)$$

This corresponds to a sparse dot product over shared vocabulary indices.

MaxSim Aggregation. For each query token t , we select the document token with maximum similarity:

$$s_{b_1,b_2,t}^* = \arg \max_{s \in \{1, \dots, T_d\}} \text{sim}(t, s). \quad (26)$$

The pairwise query–document score is then

$$S_{b_1,b_2} = \sum_{t=1}^{T_q} \text{sim}(t, s_{b_1,b_2,t}^*). \quad (27)$$

The forward pass additionally stores the argmax indices $s_{b_1,b_2,t}^*$ for use in backpropagation.

Backward Pass. Let $g_{b_1,b_2} = \frac{\partial \mathcal{L}}{\partial S_{b_1,b_2}}$ denote the upstream gradient. Since MaxSim is non-linear and sparse, gradients propagate only through the maximum document tokens.

Gradient w.r.t. Query Values. For each query token (b_1, t) and sparse entry k :

$$\frac{\partial \mathcal{L}}{\partial q_{b_1,t,k}} = g_{b_1,b_2} \sum_{\ell=1}^K \mathbb{I}[i_{b_1,t,k}^Q = i_{b_2,s^*,\ell}^D] d_{b_2,s^*,\ell}, \quad (28)$$

where $s^* = s_{b_1,b_2,t}^*$.

Gradient w.r.t. Document Values. For each document token (b_2, s^*) and sparse entry ℓ :

$$\frac{\partial \mathcal{L}}{\partial d_{b_2,s^*,\ell}} = g_{b_1,b_2} \sum_{k=1}^K \mathbb{I}[i_{b_1,t,k}^Q = i_{b_2,s^*,\ell}^D] q_{b_1,t,k}, \quad (29)$$

accumulated over all query tokens t for which s^* is the MaxSim winner.

D Ablation analysis

We conducted ablation analysis to evaluate the impact of different hyperparameters.

SPLADE While Max Pooling can result in sparser representations, it’s harder to train and highly sensitive to hyperparameters. In contrast, Mean Pooling can generally lead to more stable training, at the cost of higher non-zero entries.

How dense embeddings encode language variations is heavily affected by the underlying tokenizers (Zhu and Jurgens, 2021; Wegmann et al., 2025). For SPLADE, we also compared with mmBERT (Marone et al., 2025), a multilingual ModernBERT with over 256k vocabulary. With the same non-embedding parameters as ModernBERT, mmBERT outperforms ModernBERT-base with higher vocabulary dimensions, yet it still underperforms ModernBERT-large, with which mmBERT has roughly the same total parameters. This finding suggests that scaling up non-embedding parameters might be more effective than scaling up the embedding dimensions.

SparCol We also trained SparCol at different k . For $k \leq 256$, SparCol was not able to converge, as the gradients were too sparse for the model to learn properly. Yet increasing k beyond 1024 did not bring increased benefits, as shown in Figure 2, and will considerably slow down the training time. During inference, increasing k still brings consistent benefits, regardless of training conditions.

Dtype	Vocabulary Size	Method	Time (ms) ↓	Peak Memory (MB) ↓	Memory Reduction ↑
FP32	55K	PyTorch Baseline	470.02	14451.9	-
		Custom Sum Pooling (Ours)	282.62	1273.1	11.35×
BF16	150K	PyTorch Baseline	308.55	19480.2	-
		Custom Sum Pooling (Ours)	298.01	1691.4	11.52×

Table 17: Latency and peak GPU memory usage for sum pooling over a complete forward and backward step with batch size $B = 64$ and sequence length $T = 512$ and hidden dimension $D = 1024$, which match our training settings. The proposed Triton kernel reduces peak memory usage by over $11\times$.

Dtype	Vocabulary Size	Method	Time (ms) ↓	Peak Memory (MB) ↓	Memory Reduction ↑
FP32	55K	PyTorch Baseline	408.83	7617.2	-
		Custom Max-Pooling (Ours)	282.34	1098.5	6.93×
BF16	150K	PyTorch Baseline	235.97	10196.7	-
		Custom Max-Pooling (Ours)	255.86	1892.7	5.39×

Table 18: Latency and peak GPU memory usage for max-pooling over a complete forward and backward step with batch size $B = 64$ and sequence length $T = 512$ and hidden dimension $D = 1024$, which match our training settings. The proposed Triton kernel consistently reduces peak memory usage by $5\text{--}7\times$.

Lexicon-based models For both SPLADE-Lexicon and SPARCOL-Lexicon, we found that allowing the update of the cluster assignment delivers consistent improvement in most cases (see Table 22), except for the cross-genre authorship attribution for SPARCOL-Lexicon. Across models, over 70%-90% of the initial assigned tokens move to different clusters during training, but still around 95% of the clusters contain at least one token. This also implies that our update approach is effective even without the explicit guarantee of preserving cluster dimensions. In Table 3, it can be seen that tokens are merged with other tokens of a similar nature, such as shared character ngram, similar token variants, and similar cultural references.

E Style Probing

Style probing results on SynthSTEL for each dimension are available in Table 32, 33, 34, 35, and 36.

F Sample interpretations

Dtype	Method	Time (s) ↓	Peak Memory (MB) ↓	Memory Reduction ↑
FP32	PyTorch Baseline	0.3169	12934.5	-
	Custom Sparse (Ours)	0.4913	1778.5	7.3×
BF16	PyTorch Baseline	0.1640	6531.2	-
	Custom Sparse (Ours)	0.3030	1017.2	6.4×

Table 19: Latency and peak GPU memory usage for SPARCOL kernel of fused unembedding and top- k selection. These are complete forward and backward steps with batch size $B = 32$, sequence length $T = 512$, embedding dimension $D = 768$, vocabulary size $V = 50,000$, and $K = 512$, which match our training settings. While the proposed Triton kernel sacrifices some latency, it reduces peak memory usage by up to $7\times$ compared to a PyTorch baseline.

Dtype	Method	Time (ms) ↓	Peak Memory (MB) ↓	Memory Reduction ↑
FP32	PyTorch Baseline	1820.60	15213.3	-
	Custom MaxSim (Ours)	1241.99	9048.2	1.68×

Table 20: Comparing PyTorch baseline to custom Triton MaxSim kernel with $B = 64$, sequence length $T = 512$, vocabulary size $V = 50,000$, and $K = 1024$. The Triton kernel reduces peak memory usage and latency.

Framework	Dim / Avg.NNZ	HRS1 / English Long		HRS2 / English Medium	
		Cross	Single	Cross	Single
Pooling					
Mean	50368 / 6167	11.1	30.6	13.0	45.3
Max	50368 / 215	0.7	2.0	0.5	7.4
Vocabulary size					
ModernBERT-base	50368 / 6167	11.1	30.6	13.0	45.3
ModernBERT-large	50368 / 8192	11.1	39.0	16.8	50.4
mmBERT-base	256000 / 8192	10.8	37.8	12.0	48.4

Table 21: Ablation analysis on different hyperparameters for SPLADE-base. All numbers are MRR@20. We only ran a small model for one epoch for quick validation.

Num. Clusters	Assignment	HRS1		HRS2	
		Cross	Single	Cross	Single
SPLADE-Lexicon					
4000	Fixed	12.9	42.1	19.6	58.4
4000	Updated	13.5	43.3	20.0	60.4
SPARCOL-Lexicon					
1000	Fixed	13.7	40.7	17.2	55.7
1000	Updated	13.0	44.6	16.8	58.1

Table 22: For Lexicon-based embeddings, updating the cluster assignment during training delivers improvements in most cases. All numbers are MRR@20.

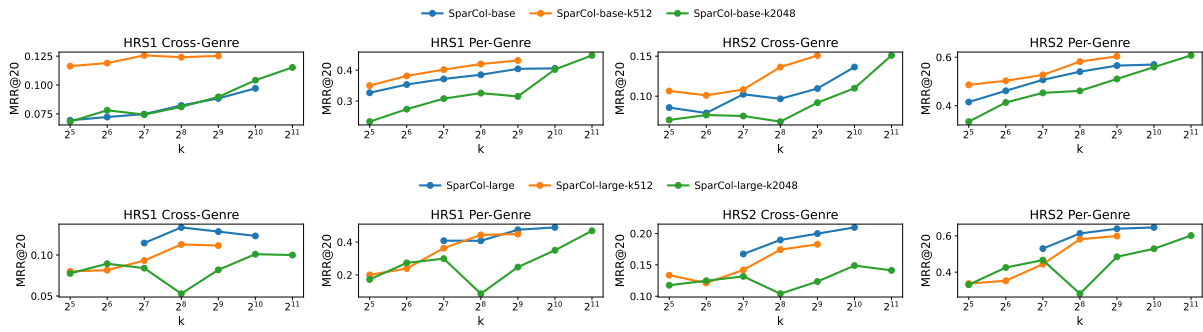


Figure 2: Training SPARCOL with different k leads to divergent performance. Increasing k to 2048 in training time did not lead to better performance at inference.

Role	Text / SPLADE Interpretation
Query	<i>The perils of no one liking you.</i> I was staff at a stylish Philadelphia restaurant for two years. The chef was a monster and treated his kitchen staff like a tyrant. Word came down that one weekend during the fall, a renowned food critic would be in to review the restaurant's new menu. The tyrant chef insisted on micromanaging every second of the weekend. Unfortunately, in his addled state of mind, he had failed to realize that he had the classic four-foot strand of toilet paper hanging from the backside of his white pants, tailing him for at least thirty minutes. No one wanted to tell him, whether it was for fear of enduring his wrath or for the humor of seeing him walking around with it hanging behind him. The incident actually made the critic's review (though she was understanding), but it seemed like a harbinger of things to come. The restaurant folded a few months later. SPLADE Top Tokens: âGL(0.742), employee(0.737), hers(0.714), .(0.664), launch(0.649), preschool(0.643), Allison(0.627), slot(0.610), \$(0.608), Cam(0.596), HR(0.586), namely(0.584), she(0.583), illa(0.575), ĉ(0.567), Hotel(0.566), Module(0.561), esse(0.558), co(0.556), retail(0.551), travel(0.548), Elle(0.543), include(0.541), optimal(0.528), Molly(0.528), map(0.526), impact(0.524), slots(0.523), ÛÛĩ(0.523), organizers(0.522), Gender(0.521), MR(0.518)
Same Author	<i>Office Sleepovers.</i> A coworker of mine named <PERSON>was swamped at work. Her boss was hard on her and she had been spread thin with so much work to do. A few months passed and I noticed that <PERSON>had become less stressed and her workload had improved dramatically. One day I had to drop something off at her office desk and went around to her side to grab a sheet of paper to write a note. What caught my eye was the reason she had been so overwhelmed: a pillow, a sleeping bag, and a toiletry bag. She had been sleeping in her office on weekends and nights to catch up. I gently scolded her and encouraged her to stop working after hours. I also coached her to apply for a new position within the company, which she did, leaving those long hours behind. SPLADE Top Tokens: meal(0.739), USDA(0.717), \$(0.710), .(0.692), Hotel(0.685), slot(0.676), retail(0.672), 's(0.667), meals(0.630), Cra(0.628), optimal(0.607), sucked(0.604), foods(0.591), Module(0.580), 're(0.572),)(0.572), Israel(0.569), tasting(0.560), travel(0.557), slots(0.551), I(0.548), illa(0.546), umble(0.544), me(0.538), Denver(0.530), Pin(0.524), Allison(0.515), laser(0.514), .(0.513), Cam(0.509), Blog(0.507), 'm(0.506)
Different Author	<i>Elderly cat won't stop sleeping in her litter box.</i> I have a sixteen-year-old long-haired female cat who lately sleeps nowhere but her litter box. She is old enough that she no longer covers her waste and often sleeps in her own filth. Sleeping in the box is not new behavior; she has done so before when stressed or ill. We recently moved to a new house, and for nearly a month she has shown no interest in resting anywhere else. I have tried treats, beds, blankets, and even converting an old box into a sleeping area. At this point I am running out of options and worry about infections. Any advice on how to stop this behavior or whether I should wait it out? SPLADE Top Tokens: 's(0.817), 're(0.768), sheep(0.732), preschool(0.730), hers(0.705),)(0.693), toddler(0.668), illa(0.657), 'm(0.635), 't(0.619), USDA(0.618), Allison(0.605), 53(0.593), aka(0.593), ...(0.583), "(0.581), ĉ(0.581), umann(0.579), co(0.569), Spencer(0.555), laser(0.554), singleton(0.548), Cra(0.543), lei(0.543), slot(0.534), HR(0.529), veterinary(0.523), dog(0.520), puppy(0.519), re-tail(0.514), animals(0.512), ä°ĩ(0.510)

Table 23: Authorship retrieval examples with SPLADE embedding interpretations. For each query, the retrieved ground-truth (GT) and a negative example, along with the top SPLADE tokens with weights.

The perils of no one liking you I was staff at a stylish Philadelphia restaurant for two years. The chef was a monster and treated his kitchen staff like a tyrant. Word came down that one weekend during the fall, a renowned food critic would be in to review the restaurant's new menu. The tyrant chef insisted on micromanaging every second of the weekend. Unfortunately, in his addled state of mind, he had failed to realize that he had the classic 4 foot strand of toilet paper hanging from the backside of his white pants, tailing him for at least 30 minutes. No one wanted to tell him, whether it was for fear of enduring his wrath or for the humor of seeing him walking around with it hanging behind him. The incident actually made the critic's review (though she was understanding) but it seemed like a harbinger of things to come. The restaurant folded a few months later.

Cluster ID	Score	Dist.	Tokens
3511	2.72	1.14	1, 2, 3, 4, 5, 19
2682	2.59	1.20	40, 32, 35, 45, 37, 36
2165	2.46	1.11	50, 60, 70, 64, 75, 65
585	2.44	1.19	t, s, o, c, p, f
2947	2.36	1.00	T, S, A, C, M, P
1148	2.34	1.34	31, 31, 33, 41
3795	2.31	0.61	", \$, ', (,), *
1893	2.24	1.16	52, 52, 51, 53
2543	2.13	0.87	90, 80, 95, 72, 99, 96
3935	2.11	0.91	200, 200, 100, 100, 500, 300
2602	1.91	0.83	a, b, c, d, e, f
390	1.84	1.29	A, B, C, D, E, F
3448	1.75	1.44	0, 1, 2, 3, 4, 5
169	1.73	0.94	awkward, shy, sore, coy, dab
2648	1.73	1.13	four, five, six, eight, seven, nine
3182	1.65	1.41	Republican, Senate, Democratic, Democrats, Republicans, GOP
3102	1.57	0.92	father, Father, father, uncle, grandmother, grandfather
1949	1.53	0.84	commit, committed, dedicated, commitment, devoted, commit
2940	1.48	0.91	pay, paid, payment, fees, fee, paying
3740	1.47	1.48	mom, mother, dad, Mother, Mom, mother
1964	1.46	0.92	qu, bu, sol, sed, ses
870	1.43	0.80	con, com, con, com, co, Com
1167	1.36	0.78	re, se, ne, Se, Re, Re
396	1.32	1.03	rare, unusual, fancy, luxury, exotic, eccentric
1959	1.31	1.08	admitted, admission, admit, admits, honestly, admissions

Table 24: Authorship retrieval query examples with SPLADE-Lexicon-1000 embedding interpretations.

Office Sleepovers A coworker of mine named <PERSON> has swamped at work. Her boss was hard on her and she had been spread thin with so much work to do. A few months pass and I notice that <PERSON> had become less stressed and her workload had improved dramatically. One day I had to drop something off at her office desk and had to go around to her side in order to grab a sheet of paper to write a note on. What caught my eye was the reason <PERSON> had gotten so caught up: a pillow, a sleeping bag, and toiletry bag. The poor woman had been sleeping in her office on weekends and nights in order to play catch-up. I gently and playfully scolded her when I saw her next and demanded she not work after hours any longer. I also coached her into applying for a new position within our company and she did, leaving those extra long hours behind her.

Cluster ID	Score	Dist.	Tokens
3511	2.74	1.14	1, 2, 3, 4, 5, 19
2682	2.57	1.20	40, 32, 35, 45, 37, 36
2947	2.44	1.00	T, S, A, C, M, P
585	2.40	1.19	t, s, o, c, p, f
1148	2.39	1.34	31, 31, 33, 41
2165	2.38	1.11	50, 60, 70, 64, 75, 65
3795	2.33	0.61	", \$, ', (,), *
1893	2.21	1.16	52, 52, 51, 53
2543	2.11	0.87	90, 80, 95, 72, 99, 96
390	2.10	1.29	A, B, C, D, E, F
3935	2.02	0.91	200, 200, 100, 100, 500, 300
3448	1.94	1.44	0, 1, 2, 3, 4, 5
169	1.90	0.94	awkward, shy, sore, coy, dab
2602	1.90	0.83	a, b, c, d, e, f
2648	1.89	1.13	four, five, six, eight, seven, nine
1800	1.75	1.13	well, Well, well, Well, wells, goodness
3875	1.71	0.92	go, going, go, went, Go, goes
1167	1.54	0.78	re, se, ne, Se, Re, Re
870	1.51	0.80	con, com, con, com, co, Com
1176	1.46	1.23	X, x, ix, x, X, Ã
2134	1.45	1.30	00, 20, 30, 50, 80, 40
1949	1.43	0.84	commit, committed, dedicated, commitment, devoted, commit
939	1.37	0.87	oci, ocial, social, ociety, society, Society
3219	1.36	0.74	bar, Bar, gate, Bar, hur, gap
377	1.36	0.69	God, demon, sin, god, sin, God

Table 25: Authorship attribution examples with SPLADE-Lexicon-1000 embedding interpretations. This text has the same Author as the query example above.

Elderly cat won't stop sleeping in her litter box I have 16 YO long-haired female cat and lately she won't sleep anywhere but in her litter box, she's also old enough that she no longer covers her poop so she's often sleeping in her own filth...(I usually cover them for her when I pass by the box, but sometimes she's on top of it before I notice). Sleeping in her box isn't exactly new behavior for her. She's done so before when she had a uti or was stressed about my sister's dog visiting. We recently moved into a new house, and while I expected her to be a little stressed in the beginning, we've now been here nearly a month and I've never seen her resting anywhere but the box. I've tried luring her into appropriate sleep spots with treats. I've tried giving her a bed and blankets to sleep on. I've tried buying a new box and converting the old hooded one into a bed (peed in immediately). At this point I'm running out of things to try...My old girl is bad at grooming herself and already prone to infections so I'm worried about her getting another uti if this keeps up. Any advice to stop my kitty from sleeping and peeing in the same place? Is there something else I can try or should I just wait it out until she's more used to the house? She hasn't displayed any other distressing symptoms that I'm aware of.			
Cluster ID	Score	Dist.	Tokens
3511	2.72	1.14	1, 2, 3, 4, 5, 19
585	2.60	1.19	t, s, o, c, p, f
2682	2.56	1.20	40, 32, 35, 45, 37, 36
1148	2.38	1.34	31, 31, 33, 41
2947	2.38	1.00	T, S, A, C, M, P
3795	2.36	0.61	", \$, ', (,), *
2165	2.33	1.11	50, 60, 70, 64, 75, 65
2602	2.14	0.83	a, b, c, d, e, f
2543	2.10	0.87	90, 80, 95, 72, 99, 96
390	2.04	1.29	A, B, C, D, E, F
3935	2.04	0.91	200, 200, 100, 100, 500, 300
3448	1.98	1.44	0, 1, 2, 3, 4, 5
169	1.97	0.94	awkward, shy, sore, coy, dab
1893	1.93	1.16	52, 52, 51, 53
1964	1.80	0.92	qu, bu, sol, sed, ses
1176	1.72	1.23	X, x, ix, x, X, Ã
3740	1.69	1.48	mom, mother, dad, Mother, Mom, mother
1959	1.64	1.08	admitted, admission, admit, admits, honestly, admissions
418	1.63	1.27	al, et, Al, Al, AL
1167	1.51	0.78	re, se, ne, Se, Re, Re
3741	1.48	1.33	wave, waves, wave, waved, Wave, waveform
2134	1.47	1.30	00, 20, 30, 50, 80, 40
168	1.39	0.95	Ann, Jane, Anne, Susan, Amy, Lucy
1994	1.37	0.98	shop, shops, Shop, shop, Gear, Cafe
624	1.32	1.49	prob, likely, likely, unlikely, apt, probable

Table 26: Authorship attribution examples with SPLADE-Lexicon-1000 embedding interpretations. This text has a different author as the query example above.

SCENARIO: Same Author
[CLS]The[4] perils of[24] no one liking you[2] [1]I[3] was[15] staff[30] at a stylish Philadelphia restaurant for two years.[12] The chef was[9] a monster and[8] treated his kitchen staff like a tyrant.[10] Word came down that one weekend during the fall,[25] a renowned food critic would be in to review the[22] restaurant's new menu.[11] The tyrant chef insisted[27] on micromanaging every second of the weekend.[13] Unfortunately, in his addled state of mind, he had[5] failed to realize that he had the classic 4 foot strand of toilet paper hanging from the backside of his white pants,[20] tailing him for at least 30 minutes.[14] No one wanted to tell him,[21] whether it was[23] for fear of enduring his wrath or[7] for the[29] humor of[28] seeing him walking around with it hanging behind him.[17] The incident actually made the critic's review (though she was[26] understanding) but[6] it seemed like a harbinger of things to come.[19] The restaurant folded a few[18] months later.[16][SEP] [CLS][4]Office Sleepovers [1,2]A cowork[30]er of[24] mine[3,15] named <PERSON> has swamped at work.[12] Her boss was[9,23,26] hard on her and[6] she had been spread thin with so much work to do. A few[18] months pass and I notice that <PERSON> had[5] become less stressed and[7] her workload had improved dramatically.[10,11] One day I had to drop something off at her office desk and had to go around to her side in order to grab a sheet of paper to write a note on.[13,14] What caught my eye was[25,28] the[22,29] reason <PERSON> had gotten so caught up: a pillow, a sleeping bag, and toiletry bag. The poor woman had been sleeping in her office on weekends and nights in order to play catch-up.[17,19] I gently and playfully scolded her when I saw her next and[8] demanded[27] she not work after hours any longer. I also coached her into applying for a new position within our company and she did,[20,21] leaving those extra long hours behind her.[16][SEP]

Table 27: Token-level MaxSim alignments of top 20 tokens between a query and a document written by the same author, as given by SPARCOL. Matching tokens are indicated with [number], whereas the number is the similarity ranking. The numbers [1] indicate tokens with the highest similarity scores.

SCENARIO: Different Author

[CLS]The[5] perils of no one liking you[2]

[1]I[6] was staff at a stylish Philadelphia restaurant for two years.[16] The chef was a monster and[8] treated his kitchen[20] staff like a tyrant.[22] Word came down that[25] one weekend during[29] the fall,[14] a renowned food critic would be in to review the[13] restaurant's[10] new menu.[15] The tyrant chef insisted on micromanaging every second of the weekend.[24] Unfortunately, in his addled state of mind, he had failed to realize that he had the classic 4 foot strand of toilet paper hanging from the backside of his white pants,[23] tailing him for at least 30 minutes.[27] No one wanted to tell him, whether it was for fear of[26] enduring his wrath or[4] for the[21] humor of[17] seeing him walking around with it hanging behind him.[19] The incident actually made the critic's[9] review ([7]though she was understanding)[12] but[3] it seemed like a harbinger of things to[30] come.[18] The restaurant folded a few months later[28].[11][SEP]

[CLS]E[5]lderly cat won't stop sleeping in her litter box

[1,2]I[6] have 16 YO long-haired female cat[20] and[16] lately she won't sleep anywhere but[3] in her litter box, she's also old enough that[14,25] she no longer covers her poop so she's often sleeping in her own filth...(I usually cover them for her when[29] I pass by the box, but sometimes she's on top of it before I notice).

Sleeping in her box isn't exactly new behavior for her. She's done so before when she had a uti or was stressed about my sister's dog visiting.[19,22,24,27] We recently moved into a new house, and while I expected her to[30] be a little stressed in the[13] beginning,[12] we've now been here nearly a month and[8] I've never seen her resting anywhere but the box. I've tried luring her into appropriate sleep spots with treats. I've tried giving her a bed and blankets to sleep on.[15,18] I've tried buying a new box and converting the old hooded one into a bed ([7]peed in immediately). At this point I'm running out of things to try...My old girl is bad at grooming herself and already prone to infections so I'm worried about[26] her getting another uti if this keeps up. Any advice to stop my kitty from sleeping and[17,23] peeing in the[21] same place? Is there something else I can try or[4] should I just wait it out until she's[9,10] more used to the house? She hasn't displayed any other distressing symptoms that I[28]'m aware of.[11][SEP]

Table 28: Token-level MaxSim alignments of top 20 tokens for a mismatched author pair, as given by SPARCOL. Matching tokens are indicated with [number], whereas the number is the similarity ranking. The numbers [1] indicate tokens with the highest similarity scores.

SCENARIO: Same Author

The[7] perils of no one liking you

[4]I[29] was staff at a stylish Philadelphia restaurant for two years. The[10] chef was a monster and[9] treated his kitchen staff like a tyrant.[18] Word[22] came down that one[23] weekend during the[3] fall,[15] a[21] renowned food critic would be in to review the restaurant's new menu.[24] The[12] tyrant chef insisted on micromanaging every second of the[1] weekend.[19] Unfortunately,[8] in his addled state of mind,[14] he had failed to realize that[11] he had the classic 4 foot strand of toilet paper hanging from the backside of his white pants,[6] tailing him for at least 30 minutes.[26] No[25] one wanted to tell him,[5] whether it was for fear of enduring his wrath or[28] for the[2] humor of seeing him walking around with it hanging behind him.[16] The[20] incident actually made the critic's review (though she was understanding) but[30] it seemed like a harbinger of things to come.[27] The[13] restaurant folded a few months later.[17]

Office[7,29] Sleepovers

[4,17]A coworker of mine named <PERSON> has swamped at work. Her[10,12,13,20,21,22,23] boss was hard on her and[9,28,30] she had been spread thin with so much work to do.[16,18,24] A few months pass and I notice that[11] <PERSON> had become less stressed and her workload had improved dramatically. One[25] day I had to drop something off at her office desk and had to go around to her side in order to grab a sheet of paper to write a note on.[19,26] What caught my eye was the reason <PERSON> had gotten so caught up: a pillow, a sleeping bag, and toiletry bag. The poor woman had been sleeping in her office on weekends and nights in order to play catch-up. I gently and playfully scolded her when I saw her next and demanded she not work after hours any longer.[27] I also coached her into applying for a new position within our company and she did.[5,6,8,14,15] leaving those extra long hours behind her.[1,2,3]

Table 29: Token-level MaxSim alignments of top 20 tokens between a query and a document written by the same author, as given by SPARCOL-Lexicon-1000. Matching tokens are indicated with [number], whereas the number is the similarity ranking. The numbers [1] indicate tokens with the highest similarity scores.

SCENARIO: Different Author

The[4] perils of no one liking you

[5][7] was staff at a stylish Philadelphia restaurant for two years. The[13] chef was a monster and[9] treated his kitchen staff like a tyrant. Word[22] came down that one[18] weekend during the[3] fall, a[17] renowned food critic would be in to review the restaurant's new menu. The[15] tyrant chef insisted on micromanaging every second of the[1] weekend.[28] Unfortunately, in his addled state of mind,[27] he had failed to realize that[23] he had the[29] classic 4 foot strand of toilet paper hanging from the backside of his white pants,[30] tailing him for at least 30 minutes. No[19] one wanted to tell him,[11] whether[20] it was for fear of enduring his wrath or[6] for the[2] humor of seeing him walking around with it hanging behind him.[24] The[16] incident actually[25] made the[26] critic's review ([12]though[21] she was understanding) but[10] it seemed like a harbinger of things to come. The[14] restaurant folded a few months later.[8]

E[4]lderly cat won't stop sleeping in her litter box

[5,8][7] have 16 YO long-haired female cat and lately she won't sleep anywhere but in her litter box, she's also[25] old enough that[23] she no longer covers her poop so she's often sleeping in her own filth...(I usually cover them for her when I pass by the box, but sometimes she's on top of it before I notice).

Sleep[13,14,15,16,26]ing in her box isn't exactly new behavior for her. She[22]'s done so before when she had a uti or was stressed about my sister's dog visiting. We recently moved into a new house,[11,30] and[20] while I expected her to be a little stressed in the[1] beginning,[27] we've now been here nearly a month and[9] I've never seen her resting anywhere but the box. I've tried luring her into appropriate sleep spots with treats. I've tried giving her a bed and blankets to sleep on. I've tried buying a new box and converting the[2,3] old hooded one into a bed ([12]peed in immediately). At this point I'm running out of things to try...[21]My[17,18,19,29] old girl is bad at grooming herself and already prone to infections so I'm worried about her getting another uti if this keeps up.[24,28] Any advice to stop my kitty from sleeping and peeing in the same place? Is there something else I can try or[6,10] should I just wait it out until she's more used to the house? She hasn't displayed any other distressing symptoms that I'm aware of.

Table 30: Token-level MaxSim alignments of top 20 tokens for a mismatched author pair, as given by SPARCOL-Lexicon-1000. Matching tokens are indicated with [number], whereas the number is the similarity ranking. The numbers [1] indicate tokens with the highest similarity scores.

Token	Top-8 Logits
[CLS]	[CLS] (1.612), [MASK] (1.543), [SEP] (1.498), "... (1.484), "? (1.480), ". (1.478), '. (1.471), ". (1.464)
The	Financial (1.259), =" (1.257), == (1.251), _va (1.247), Dear (1.240), vae (1.238), "... (1.238), — (1.227)
_per	_per (1.148), _Per (1.134), Per (1.097), _va (1.091), _Gra (1.085), _vag (1.076), _fail (1.061), _imper (1.058)
ils	ils (1.256), _hazards (1.167), _peril (1.166), _virtues (1.166), _dangers (1.165), vir (1.152), _benefits (1.143), _horrors (1.141)
_of	_of (1.055), \n (1.033), _“ (1.022), [CLS] (1.008), _Of (1.001), _‘ (0.970), _" (0.969), [MASK] (0.965)
_no	_“ (0.974), \n (0.966), “ (0.932), _‘ (0.900), ’ (0.868), _no (0.863), _" (0.862), ... (0.858)
_one	_“ (0.955), \n (0.939), ’ (0.911), “ (0.905), ... (0.897), _" (0.886), ". (0.876), ". (0.868)
_liking	_liking (1.051), \n (1.036), “ (0.998), ... (0.991), [CLS] (0.989), . (0.985), ... (0.979), _loving (0.972)
_you	\n (1.225), ". (1.162), ". (1.159), [CLS] (1.154), [MASK] (1.150), ... (1.150), '. (1.140), ", (1.132)
\n	". (1.274), \n (1.271), "... (1.269), ’? (1.248), "? (1.248), : (1.243), . (1.243), "... (1.240)
I	I (1.226), mie (1.215), _I (1.201), _heme (1.197), annie (1.195), Î (1.191), QI (1.187), medi (1.177)
_was	_flew (1.201), _was (1.166), _Was (1.154), Was (1.154), was (1.149), _Air (1.149), _Fly (1.137), annie (1.133)
_staff	_staff (1.218), _yeast (1.181), _Staff (1.175), staff (1.154), Staff (1.145), yll (1.144), _assistant (1.131), _assistants (1.125)

Table 31: SparCol token-level sparse representations. Tokens are rendered in a human-readable form: ‘_’ denotes a leading whitespace and ‘\n’ denotes a newline.

Style Feature	SPLADE-v3 (semantic)	SPLADE-Base	SPLADE-Large	SPLADE-Lexicon-Large
Active / Passive	57.2 / 64.5 role specialist invented you [SEP]	81.0 / 89.7 stimulated enter — Sid hereinafter	82.2 / 89.1 bu ba be belonging Dom	73.8 / 82.0 cluster_3618: frequency Hz avelength wavelength frequencies requency Hz frequency vibration wavelengths
Affective / Perceptual Process	75.8 / 85.0 notice excitement feel excited see	75.8 / 85.3 ridge hero reflection Joseph	74.2 / 81.3 input excited target influence Many	72.7 / 80.2 cluster_1220: feel felt feeling Fe feels fel Feel felt feel
Affective / Cognitive Processes	77.2 / 84.2 feel understand strategy excited feeling	73.0 / 84.2 dc couples steam HR hero	73.5 / 79.0 Will research Alger structure Bon	76.0 / 84.5 cluster_1220: feel felt feeling Fe feels fel Feel felt feel
All Lower Case / Proper Capitalization	50.0 / 50.0 [PAD] strolled biotechnology spines sharif	95.8 / 99.9 Both Move Plus Many Parent	97.7 / 100.0 <> <newline:1> super SU i	95.8 / 99.6 cluster_3161: guide guidance primer Guide primers guided guide guides guiding guided
All Upper Case / Proper Capitalization	50.0 / 50.0 [PAD] strolled biotechnology spines sharif	100.0 / 100.0 ITY FLAG FUN TRAN LET	100.0 / 100.0 SU THE FOR ONE OO	100.0 / 100.0 cluster_1111: LE IL NE BE NE LE
Certain / Uncertain	60.7 / 67.9 definitely surely success certainly clearly	75.0 / 85.2 may Maybe sincere May plic	74.8 / 82.2 maybe Or possibly ? mid	71.5 / 79.8 cluster_3785: may might May May may might Might MAY Maya
Cognitive / Perceptual Process	70.2 / 78.1 observe analyze understand notice observed	63.7 / 71.7 analyzed Rainbow watch ocar flawed	62.2 / 68.6 birth objects ## visible founder	60.7 / 68.0 cluster_2432: observ observed obser observations observation observe Observ observing observational observ
Complex / Simple	55.2 / 57.9 [SEP] improve institute tourism app	77.8 / 85.2 co reflection prevalent Constitution hers	74.8 / 83.3 thereby — sister approximately sham	75.0 / 82.2 cluster_252: entity entities Entity entity Entity

Table 32: SynthSTEL linear-probing results by individual style feature. Each cell reports accuracy / AUC, macro-averaged over 10 pair-level 80/20 train/test splits. (part 1 of 5).

Style Feature	SPLADE-v3 (semantic)	SPLADE-Base	SPLADE-Large	SPLADE-Lexicon-Large
Fluent / Disfluent Sentence	51.0 / 52.0 thrill diet role prevention company	84.2 / 90.9 852 <space:1,tab:1> fog preschool rx	84.5 / 92.5 , ful 99 bus 18	82.0 / 89.1 cluster_3685: face face faces facing faced confront pose facial posed poses
Formal / Informal	66.2 / 76.0 gonna no mascot future really	94.0 / 99.1 sucked really Tot moiety deceased	95.8 / 99.4 approximately viewed desired containing dad	95.0 / 99.2 cluster_252: entity entities Entity entity Entity
Long / Short Average Word Length	79.5 / 87.2 comprehensive [SEP] big no wow	99.8 / 100.0 weather moiety interventions systemic prevalent	96.8 / 99.9 deemed approximately prior dad various	98.8 / 100.0 cluster_252: entity entities Entity entity Entity
Offensive / Non-Offensive	74.8 / 84.6 [SEP] stupid annoying pathetic celebrity	91.0 / 95.3 sucked shallow junk - —	83.8 / 90.9 kill ugly Based Due anal	82.2 / 90.5 cluster_3444: fuck shit cum screw fucking damn bitch piss shit Fuck
Polite / Impolite	72.8 / 79.3 scandal no proposed toy request	90.0 / 97.1 sucked deceased really ' rocks	91.3 / 96.8 ugly deemed Due 240 alter	87.8 / 96.0 cluster_122: interesting odd strange unusual weird odd curious peculiar strang Strange
Positive / Negative	72.7 / 79.9 benefit no scandal [SEP] disadvantage	75.8 / 86.8 NY If Wang AC fog	76.8 / 82.8 failure afraid evil adin ful	72.7 / 80.5 cluster_1494: amazing wonderful exciting awesome remarkable brilli impressive incredible fantastic brilliant
Present / Future Focused	68.0 / 78.8 currently future will 2020 tomorrow	68.7 / 77.6 will Calculate Will will Will	71.5 / 78.6 future will fall lying affect	66.5 / 71.9 cluster_2137: uture future fut Future Future future uristic futures Fut futuro
Present / Past Focused	75.2 / 83.8 currently yesterday no invented did	87.2 / 94.2 had resulted consisted arrived contended	72.5 / 79.1 failure years achieved is af	67.2 / 74.2 cluster_1917: mem memory mem Mem Mem memories memor MEM memory Memorial

Table 33: SynthSTEL linear-probing results by individual style feature. Each cell reports accuracy / AUC, macro-averaged over 10 pair-level 80/20 train/test splits. (part 2 of 5).

Style Feature	SPLADE-v3 (semantic)	SPLADE-Base	SPLADE-Large	SPLADE-Lexicon-Large
Self / Audience Focused	69.5 / 83.0 their they my i no	96.5 / 99.9 I me my I My	90.0 / 94.6 their My me themselves i	89.0 / 95.2 cluster_2752: us me my me my My Me My Me Us
Self / Inclusive Focused	75.2 / 85.0 our we his he she	94.8 / 99.4 we us me our I	86.2 / 94.0 we ourselves Our My Our	76.5 / 83.2 cluster_2752: us me my me my My Me My Me Us
Self / Third-Person Singular	68.7 / 82.0 his he she her no	87.5 / 93.3 he I she me hers	79.3 / 85.0 He He his she Your	76.0 / 83.5 cluster_1023: he he his her we she He him We We
Self / You Focused	69.0 / 76.9 you his no your he	91.2 / 95.3 your me I Your You	85.8 / 92.8 yourself Your You You Your	76.5 / 88.4 cluster_2752: us me my me my My Me My Me Us
Misspelled / Normal Sentence	78.8 / 90.3 [SEP] benefit dragon opportunity yu	98.2 / 99.7 rx saint National - Ø	97.5 / 99.2 ou sam IP hyp adv	96.2 / 99.0 cluster_420:) , - . / : ; _ o s
Text Emojis / No Emojis	62.0 / 68.0 word d = ##d equation	98.5 / 99.9 (+) DD <tab:1> Rainbow MR	95.5 / 99.0 .= d SU (= X	95.8 / 98.8 cluster_1757: music dance Music musical jazz Music Dance music Bach Jazz
With / Less Frequent Articles	52.8 / 56.6 opportunity toy no restaurant game	87.2 / 94.5 the Alberta Joseph \u0093 ?	81.5 / 89.9 the goal THE plus it	83.0 / 89.4 cluster_2381: The The the THE FOR THE
With / Less Frequent Common Verbs	62.0 / 66.4 [SEP] no ritual specialist regulation	89.5 / 96.1 moiety adequate film maneuver furnish	88.8 / 95.6 alter alleged deemed accused desired	89.8 / 97.1 cluster_252: entity entities Entity entity Entity

Table 34: SynthSTEL linear-probing results by individual style feature. Each cell reports accuracy / AUC, macro-averaged over 10 pair-level 80/20 train/test splits. (part 3 of 5).

Style Feature	SPLADE-v3 (semantic)	SPLADE-Base	SPLADE-Large	SPLADE-Lexicon-Large
With / Less Frequent Conjunctions	56.2 / 56.7 no physics step latest facility	78.5 / 87.9 also q and and ;	83.0 / 88.6 ve Force DE ia choice	75.5 / 85.6 cluster_2277: ip ips IP IP ip ping ips Ip
With / Without Contractions	52.5 / 56.3 timothy will no role intensive	96.5 / 99.5 're 's 't local	91.2 / 97.4 'm +. ' plus 've	87.0 / 95.5 cluster_114: " \$ ' I \ ' d i t e d
With / Less Frequent Determiners	55.2 / 57.6 future latest [SEP] quote biggest	82.0 / 92.1 the Share deserved Samsung King	80.7 / 87.0 coach constant approximately THE AG	76.2 / 84.7 cluster_3804: find found findings finding Found find finds Find Find found
With / Less Frequent Digits	71.5 / 79.5 2020 many no % 202	84.2 / 92.3 2021 plic Many 50 schema	74.2 / 83.5 ent ourselves 2021 EL act	79.0 / 85.6 cluster_3011: cov OVID avirus COVID Brexit coronavirus pandemic CoV SARS Cov
With Emojis / No Emojis	58.8 / 65.3 [UNK] [SEP] word flavor dutch	100.0 / 100.0 Ĭ face singleton padd pun	98.0 / 99.9 ā seq	98.2 / 99.8 cluster_2818: ĭ ġ £ ¥ ¢ ³ ¼ ½ G ġ
With / Less Frequent Punctuation	54.2 / 57.5 [SEP] who quote location comfort	85.0 / 95.1 , ; - \ co	90.0 / 96.6 -, Dom dependent clock \$,	88.5 / 95.0 cluster_1436: vo voice vo voices Vo voice VO Vo Voice VO
With / Less Frequent Function Words	58.0 / 58.7 no your you specialist were	85.5 / 94.0 Plus namely Tot semantic mil	82.8 / 91.4 approximately bait Ocean star Art	80.5 / 91.3 cluster_1030: camera caption eras cameras Getty Camera Camera camera caption shutter
With / Without Humor	73.8 / 78.8 [SEP] mascot slogan cartoon toy	90.8 / 95.2 Tot film Cra bob optimal	84.5 / 92.8 Let pull hole About research	83.5 / 92.0 cluster_420:) , - . / : ! _ o s

Table 35: SynthSTEL linear-probing results by individual style feature. Each cell reports accuracy / AUC, macro-averaged over 10 pair-level 80/20 train/test splits. (part 4 of 5).

Style Feature	SPLADE-v3 (semantic)	SPLADE-Base	SPLADE-Large	SPLADE-Lexicon-Large
With / Without Metaphor	70.2 / 78.8 [SEP] no metaphor fortress mascot	87.5 / 95.0 travel Han summary notification bob	87.0 / 94.3 Review sea approximately correctly cloth	84.8 / 93.6 cluster_1030: camera caption eras cameras Getty Camera Camera camera caption shutter
With / Without Nominalizations	63.5 / 69.7 achievement committee [SEP] you scandal	82.5 / 90.8 ? lesion involvement ' -	79.8 / 89.2 traditional latter Ĥ approximately Jordan	75.0 / 80.5 cluster_3766: consistency clarity continuity persistence fragmentation resilience
With / Without Number Substitution	67.3 / 77.3 [SEP] ##3 no who 4	87.5 / 93.0 2 <space:1,tab:1> rx 2 bow	84.5 / 92.5 SU 60 super BU P	84.0 / 90.3 cluster_2450: 0 1 2 1 2 0 3 4 5
With / Less Frequent Personal Pronouns	67.5 / 77.6 your no you specialist his	82.5 / 92.8 your we We I 852	81.5 / 89.6 you pending 240 father You	79.0 / 87.1 cluster_1578: our your our ours Our Your Our Your oured OUR
With / Less Frequent Prepositions	54.5 / 57.2 no [SEP] character affect football	79.8 / 86.6 {}, the , - Lemma	75.0 / 82.7 Att Ac ba -, material	72.0 / 81.3 cluster_1436: vo voice vo voices Vo voice VO Vo Voice VO
With / Less Frequent Pronouns	63.5 / 73.0 you your no their his	89.0 / 94.9 proper we Israel your organizers	85.8 / 94.3 240 lone You Anti offender	82.0 / 90.2 cluster_1578: our your our ours Our Your Our Your oured OUR
With / Without Sarcasm	62.8 / 68.1 no [SEP] oh joy wow	79.2 / 87.9 , iber { }, ! faculty	82.5 / 90.6 ol hyp atom ap films	80.5 / 86.7 cluster_388: ight exact perfect exactly straight perfectly precisely precise actly Exactly
With / Without Uppercase Letters	50.0 / 50.0 [PAD] strolled biotechnology spines sharif	98.0 / 99.7 Move Split President Calculate Flow	98.0 / 99.7 <> <newline:1> the SU i	96.0 / 99.8 cluster_252: entity entities Entity entity Entity

Table 36: SynthSTEL linear-probing results by individual style feature. Each cell reports accuracy / AUC, macro-averaged over 10 pair-level 80/20 train/test splits. (part 5 of 5).