Efficient Document Embeddings via Self-Contrastive Bregman Divergence Learning

Daniel Saggau*   **Mina Rezaei***   **Bernd Bischl**
Department of Statistics, Ludwig Maximilian University of Munich (LMU), Germany
Munich Center for Machine Learning (MCML), Germany

Ilias Chalkidis
Department of Computer Science, University of Copenhagen, Denmark

Abstract

Learning quality document embeddings is a fundamental problem in natural language processing (NLP), information retrieval (IR), recommendation systems, and search engines. Despite recent advances in the development of transformer-based models that produce sentence embeddings with self-contrastive learning, the encoding of long documents (Ks of words) is still challenging with respect to both efficiency and quality considerations. Therefore, we train Longfomer-based document encoders using a state-of-the-art unsupervised contrastive learning method (SimCSE). Further on, we complement the baseline method - siamese neural network- with additional convex neural networks based on functional Bregman divergence aiming to enhance the quality of the output document representations. We show that overall the combination of a self-contrastive siamese network and our proposed neural Bregman network outperforms the baselines in two linear classification settings on three long document topic classification tasks from the legal and biomedical domains.

1 Introduction

The development of quality document encoders is of paramount importance for several NLP applications, such as long document classification tasks with biomedical (Johnson et al., 2016), or legal (Chalkidis et al., 2022b) documents, as well as information retrieval tasks (Chalkidis et al., 2021a; Rabelo et al., 2022; Nentidis et al., 2022). Despite the recent advances in the development of transformer-based sentence encoders (Reimers and Gurevych, 2019; Gao et al., 2021; Liu et al., 2021; Klein and Nabi, 2022a) via unsupervised contrastive learning, little do we know about the potential of neural document-level encoders targeting the encoding of long documents (Ks of words).

---

*The authors contributed equally to this work.

Training Corpus	Average Text Length
Reimers and Gurevych (2019) *inter alia*	
SNLI	22
MNLI	113
MS Marco	335
Wikipedia	200
Our Work	
ECtHR	1,613
MIMIC	1,621
SCOTUS	5,853

Table 1: Text length across corpora that have been used for self-contrastive pre-training in the NLP literature.

The computational complexity of standard Transformer-based models (Vaswani et al., 2017; Devlin et al., 2019) (PLMs) given the quadratic self-attention operations poses challenges in encoding long documents. To address this computational problem, researchers have introduced efficient sparse attention networks, such as Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020), and Hierarchical Transformers (Chalkidis et al., 2022a). Nonetheless, fine-tuning such models in downstream tasks is computationally expensive; hence we need to develop efficient document encoders that produce quality document representations that can be used for downstream tasks out-of-the-box, i.e., without fully (end-to-end) fine-tuning the pre-trained encoder, if not at all.

Besides computational complexity, building good representation models for encoding long documents can be challenging due to document length. Long documents contain more information than shorter documents, making it more difficult to capture all the relevant information in a fixed-size representation. In addition, long documents may have sections with different topics, which increases the complexity of encoding that usually leads to collapsing representations (Jing et al., 2022). More-

over, long documents can be semantically incoherent, meaning that content may not be logically related or may contain irrelevant information. For these reasons, it is challenging to create a quality representation that captures the most important information in the document.

To the best of our knowledge, we are the first to explore the application of self-contrastive learning for long documents (Table 1). The contributions of our work are threefold:

(i) We train Longformer-based document encoders using a state-of-the-art self-contrastive learning method, SimCSE by Gao et al. (2021).

(ii) We further enhance the quality of the latent representations using convex neural networks based on functional Bregman divergence. The network is optimized based on self-contrastive loss with divergence loss functions (Rezaei et al., 2021).

(iii) We perform extensive experiments to highlight the empirical benefits of learning representation using unsupervised contrastive and our proposed enhanced self-contrastive divergence loss. We compare our method with baselines on three long document topic classification tasks from the legal and biomedical domain.

2   Related Work

Document Encoders   The need for quality document representations has always been an active topic of NLP research. Initial work on statistical NLP focused on representing documents as Bag of Words (BoW), in which direction TF-IDF representations were the standard for a long time. In the early days of deep learning in NLP, models developed to represent words with latent representations, such as Word2Vec (Mikolov et al., 2013), and GloVe (Pennington et al., 2014). Within this research domain, the use of word embedding centroids as document embeddings, and the development of the Doc2Vec (Le and Mikolov, 2014) model were proposed. Given the advanced compute needs to encode documents with neural networks, follow-up work mainly developed around sentence/paragraph-level representations, such as Skip Thoughts of Kiros et al. (2015), which relies on an RNN encoder. In the era of pre-trained Transformer-based language models, Reimers and Gurevych (2019) proposed the Sentence Transformers framework in order to develop quality dense sentence representations. Many

works followed a similar direction relying on a self-supervised contrastive learning setup, where most ideas are adopted mainly from Computer Vision literature (Chen et al., 2020; Bardes et al., 2022).

Self-Supervised Contrastive Learning in NLP Several self-contrastive methods have been proposed so far for NLP applications. To name a few: MirrorRoBERTa (Liu et al., 2021), SCD (Klein and Nabi, 2022b), miCSE (Klein and Nabi, 2022a), DeCluTR (Giorgi et al., 2021), and SimCSE (Gao et al., 2021) – described in Section 3.2–, all create augmented versions (views) of the original sentences using varying dropout and comparing their similarity. The application of such methods is limited to short sentences and relevant downstream tasks, e.g., sentence similarity, while these methods do not use any additional component to maximize diversity in latent feature representations.

3   Methods

3.1   Base Model - Longformer

We experiment with Longformer (Beltagy et al., 2020), a well-known and relatively simple sparse-attention Transformer. Longformer uses two sets of attention, namely sliding window attention and global attention. Instead of using the full attention mechanism, the sliding-window attention gives local context higher importance. Given a fixed window size w, each token attends to $\frac{1}{2}w$ tokens on the respective side. The required memory for this is $O(n \times w)$. Sliding-window attention is combined with global attention from/to the [CLS] token.

Domain-Adapted Longformer: As a baseline, we use Longformer$_{DA}$ models which are Longformer models warm-started from domain-specific PLMs. To do so, we clone the original positional embeddings 8× to encode sequences up to 4096 tokens. The rest of the parameters (word embeddings, transformers layers) can be directly transferred, with the exception of Longformer's global attention K, Q, V matrices, which we warm-start from the standard (local) attention matrices, following Beltagy et al. (2020). All parameters are updated during training.

For legal applications (Section 4.1), we warm-start our models from Legal-BERT (Chalkidis et al., 2020), a BERT model pre-trained on diverse English legal corpora, while for the biomedical one, we use BioBERT (Lee et al., 2020), a BERT model pre-trained on biomedical corpora.
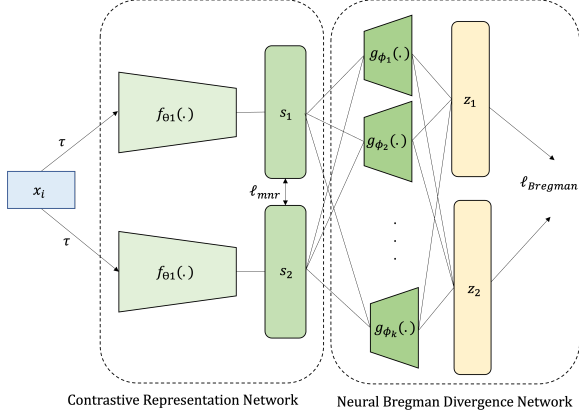
Figure 1: Illustration of our proposed self-contrastive method combining SimCSE of Gao et al. (2021) (left part) with the additional Bregman divergence networks and objective of Rezaei et al. (2021) (right part).

3.2 Self-supervised Contrastive Learning

To use our Longformer$_{DA}$ for self-supervised contrastive learning, we need to use a Siamese network architecture (left part of Figure 1). Assume we have mini-batch $\mathcal{D} = \{(x_i)\}_{i=1}^N$ of N documents. As positive pairs (x_i, x_i^+), the method uses augmented (noised) versions of the input feature x_i. As negative pairs (x_i, x_i^-), all remaining N-1 documents in a mini-batch are used. The augmentations take place in the encoder block f_θ of the model. θ is the parameterization of the encoder. We use the SimCSE (Gao et al., 2021) framework, in which case the encoder f_θ is a pre-trained language model, Longformer$_{DA}$ in our case, and augmentation comes in the form of varying token dropout (masking) rate (τ). The loss objective used in the unsupervised version of SimCSE is the multiple negatives ranking loss (ℓ_{mnr}):

$$\ell_{mnr} = -\frac{1}{n} \sum_{i=1}^n \frac{\exp\left(f\left(s_i, \tilde{s}_i\right)\right)}{\sum_j \exp\left(f\left(s_i, s_j\right)\right)} \quad (1)$$

where $\tilde{s}_i$ is the positive augmented input sequence in the mini-batch, and $\tilde{s}_j$ are the negatives. Multiple negatives ranking loss takes a pair of representations $(s_i, \tilde{s}_i)$ and compares these with negative samples in a mini-batch. In our experiments, we train such models, dubbed Longformer$_{DA+SimCSE}$.

3.2.1 Bregman Divergence Loss

We complement this method with an additional ensemble of subnetworks optimized by functional Bregman divergence aiming to improve the output document latent representations further. Specifically, the embedding of self-contrastive networks

further passes to k-independent subnetworks to promote diversity in feature representations.

The s_i and s_j vectors from the contrastive framework are mapped to k-independent ensemble of neural networks that are optimized using functional Bregman divergence.

$$G_\phi(s_a, s_b) = \phi(s_a) - \phi(s_b) -$$
$$\int [s_a(x) - s_b(x)]\delta\phi(s_b)(x)d\mu(x) \quad (2)$$

s_a and s_b are vectors output by the self-contrastive network, and ϕ is a strictly convex function and can be described via a linear functional, consisting of weights w_k and biases ϵ_k. The function $\phi(s_a)$ is approximate by:

$$\phi(s_a) = \sup_{(w,\epsilon_w)\in Q} \int s_a(x)w(x)dx + \epsilon_w \quad (3)$$

We take the empirical distribution of the projection representation to compute $\hat{s}_a$ and $\hat{s}_b$. Specifically we define: $\hat{s}_i = \text{argmax}_k[\int s_a(x)w_k(x)dx + \epsilon_k]$ for i = (a,b). Using the above specification and $\phi(s_a)$, we get the following functional divergence term:

$$G(s_a, s_b) = (\int s_a(x)w_{\hat{s}_a}(x)dx + \epsilon_{\hat{s}_a} -$$
$$(\int s_a(x)w_{\hat{s}_b}(x)dx + \epsilon_{\hat{s}_b}) \quad (4)$$

Each sub-network produces a separate output (right part of Figure 1). The divergence is then computed using the output at point $\hat{s}_a$ and $\hat{s}_b$ using the projections as input. We convert the divergence to similarity using a Gaussian kernel as done by Rezaei et al. (2021).[1]

$$\psi = \exp\left(-G/2\sigma^2\right) \quad (5)$$

The mini-batch has size N. For empirical distributions $s_a\alpha(z_i)$, $s_b(z_j)$ where i and j are the respective index for the two branches and z the projector representation, we have:

$$\ell_{\text{Bregman}(s_a(z_i),s_b(z_j))} = \frac{-\log(\exp\left(\psi_{i,j}\right)}{\sum_{t=1}^N \exp\left(\psi_{i,k}\right))} \quad (6)$$

The final objective function is computed on the combination of as follows:

$$L_{\text{Total}} = \ell_{mnr} + \lambda \cdot \ell_{\text{Bregman}} \quad (7)$$

---
[1] Rezaei et al. (2021) explore various monotone transformations. The Gaussian kernel performed best compared to other transformation methods.

| Method | ECtHR | | SCOTUS | | MIMIC | | Avg. | Training Efficiency | |
	μ-F$_1$	m-F$_1$	μ-F$_1$	m-F$_1$	μ-F$_1$	m-F$_1$	μ-F$_1$	Time (h)	Params (%)
DOCUMENT EMBEDDING + MLP									
Longformer$_{DA}$	61.4	47.8	65.7	50.5	63.9	48.3	63.6	4.5h	0.5%
Longformer$_{DA+SimCSE}$	<u>64.4</u>	<u>55.0</u>	<u>69.2</u>	<u>57.5</u>	<u>66.0</u>	**52.9**	<u>66.5</u>	»	»
Longformer$_{DA+SimCSE+Bregman}$	**64.8**	**56.3**	**69.7**	**58.8**	**66.7**	<u>51.7</u>	**67.1**	»	»
DOCUMENT EMBEDDING + LINEAR LAYER									
Longformer$_{DA}$	**73.7**	**62.4**	69.3	59.0	<u>59.4</u>	21.7	<u>67.5</u>	1h	0.5%
Longformer$_{DA+SimCSE}$	70.6	56.2	<u>69.6</u>	<u>60.9</u>	59.2	**23.0**	66.5	»	»
Longformer$_{DA+SimCSE+Bregman}$	<u>73.3</u>	<u>59.5</u>	**71.4**	**62.0**	**59.6**	<u>22.7</u>	**68.1**	»	»
END-TO-END FINE-TUNING (CEILING)									
Longformer$_{DA}$	78.8	71.5	75.2	63.2	78.9	56.4	77.6	8h	100%

Table 2: Test Results for all methods across all datasets. Best performance in **bold**, and second-best score is <u>underlined</u>. We also report average training time and the percentage of parameters that are trainable.

Where λ is a scalar hyperparameter to weigh the relative importance of the Bregman divergence and contrastive loss. In our experiments, we train such models, dubbed Longformer$_{DA+SimCSE+Bregman}$.

4 Experimental Set-up

4.1 Datasets and Tasks

ECtHR (Chalkidis et al., 2021b) dataset contains 11k cases from the European Court of Human Rights (ECtHR). This is a multi-label topic classification task, where given the facts of an ECtHR case, the model has to predict the alleged violated ECtHR article among ten such articles (labels).

SCOTUS (Chalkidis et al., 2022b) dataset contains 4.7k cases from the Supreme Court of US (SCOTUS). This is a single-label multi-class topic classification task, where given a SCOTUS opinion, the model has to predict the relevant area among 14 issue areas (labels).

MIMIC (Johnson et al., 2016) dataset contains approx. 50k discharge summaries from US hospitals. Each summary is annotated with one or more codes (labels) from the ICD-9 hierarchy, which has 8 levels in total. We use the 1st level of ICD-9, including 19 categories, respectively. This is a multi-label topic classification task, where given the discharge summary, the model has to predict the relevant ICD-9 top-level codes (labels).

4.2 Experimental Settings

To get insights into the quality of the learned representations out-of-the-box, we train classifiers using document embeddings as fixed (frozen) feature representations. We consider two linear classification settings: (i) Linear evaluation plugging a MLP classification head on top of the document embeddings; (ii) Linear evaluation plugging a linear classifier on top of the document embeddings.

5 Results and Discussion

In Table 2, we present the results for all examined Longformer variants across the three examined datasets and two settings using macro-F1 (m-F$_1$) and micro-F1 (μ-F$_1$) scores.

Classification performance: In the last line of Table 2, we present the results for the baseline Longformer$_{DA}$ model fine-tuned end-to-end, which is a 'ceiling' for the expected performance, comparing to the two examined linear settings, where the document encoders are not updated. We observe that in the SCOTUS dataset training models with an MLP head are really close to the ceiling performance (approx. 1-4p.p. less in μ-F$_1$). The gap is smaller for both models trained with the self-contrastive objective (+SimCSE, +SimCSE+Bregman), especially the one with the additional Bregman divergence loss, where the performance decrease in μ-F$_1$ is only 1 p.p.

In the other two datasets (ECtHR and MIMIC), the performance of the linear models is still approx. 10-15 p.p. behind the ceilings in μ-F$_1$. In ECtHR, we find that self-contrastive learning improves performance in the first settings by 3 p.p. in μ-F$_1$, while the additional divergence Bregman loss does not really improve performance. This is not the

Model	μ-F$_1$	m-F$_1$
Longformer$_{DA}$	<u>54.9</u>	<u>48.1</u>
» + SimCSE	51.8	43.6
» + SimCSE + Bregman	**56.9**	**48.5**

Table 3: Test Results for all Longformer variants for SCOTUS. Best performance in **bold**, and second-best score is <u>underlined</u>.

case, in the second linear setting (second group in Table 2), where the baseline outperforms both models. Similarly in MIMIC, we observe that self-contrastive learning improves performance in the first settings by 3 p.p. in μ-F$_1$, but the performance is comparable given linear classifiers. Overall, our enhanced self-contrastive method leads to the best results compared to its counterparts.

In Table 3, we also present results on SCOTUS in a few-shot setting using the SetFit (Tunstall et al., 2022) framework, where Bregman divergence loss improves performance compared to the baselines.

Given the overall results, we conclude that building subnetwork ensembles on top of the document embeddings can be a useful technique for encoding long documents and can help avoid the problem of collapsing representations, where the model is unable to capture all the relevant information in the input. Our approach has several advantages for long-document processing:

Efficiency considerations: In Table 2, we observe that in both linear settings where fixed document representations are used, the training time is 2-8× decreased compared to end-to-end fine-tuning, while approx. 0.5% of the parameters are trainable across cases, which directly affects the compute budget. We provide further information on the size of the models in Appendix B.

Avoidance of collapsing representations: When processing long documents, there is a risk that the representation will collapse (Jing et al., 2022), meaning that the model will not be able to capture all the relevant information in the input. By mapping the document embedding from the base encoder into smaller sub-networks, the risk of collapsing representations is reduced, as the divergence loss attempts to reduce redundancy in the feature representation by minimizing the correlation. The results shown in Table 3 in a low-resource setting further highlight the advantage of training a Longformer with contrastive divergence learning.

6 Conclusions and Future Work

We proposed and examined self-supervised contrastive divergence learning for learning representation of long documents. Our proposed method is composed of a self-contrastive learning framework followed by an ensemble of neural networks that are optimized by functional Bregman divergence. Our method showed improvement compared to the baselines on three long document topic classifications in the legal and biomedical domains, while the improvement is more vibrant in a few-shot learning setting. In future work, we would like to further investigate the impact of the Bregman divergence loss in more classification datasets and other NLP tasks, e.g., document retrieval.

Limitations

In this work, we focus on small and medium size models (up to 134M parameters), while recent work in Large Language Models (LLMs) targets models with billions of parameters(Brown et al., 2020; Chowdhery et al., 2022). It is unclear how well the performance improvement from the examined network architecture would translate to other model sizes or baseline architectures, e.g., GPT models.

Further on, it is unclear how these findings may translate to other application domains and datasets, or impact other NLP tasks, such as document retrieval/ranking. We will investigate these directions in future work.

Acknowledgments

References

Adrien Bardes, Jean Ponce, and Yann Lecun. 2022. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR 2022-10th International Conference on Learning Representations*.

---

[2]https://innovationsfonden.dk/en

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022a. An exploration of hierarchical attention transformers for efficient long document classification.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Nikolaos Manginas, Eva Katakalou, and Prodromos Malakasiotis. 2021a. Regulatory compliance through Doc2Doc information retrieval: A case study in EU/UK legislation where text similarity has limitations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3498–3511, Online. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021b. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 226–241, Online. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022b. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. Declutr: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2022. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, 28.

Tassilo Klein and Moin Nabi. 2022a. micse: Mutual information contrastive learning for low-shot sentence embeddings. *arXiv preprint arXiv:2211.04928*.

Tassilo Klein and Moin Nabi. 2022b. SCD: Self-contrastive decorrelation of sentence embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 394–400, Dublin, Ireland. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1459, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Scottsdale, AZ.

Anastasios Nentidis, Georgios Katsimpras, Eirini Vandorou, Anastasia Krithara, Antonio Miranda-Escalada, Luis Gasco, Martin Krallinger, and Georgios Paliouras. 2022. Overview of bioasq 2022: The tenth bioasq challenge on large-scale biomedical semantic indexing and question answering. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 13th International Conference of the CLEF Association, CLEF 2022, Bologna, Italy, September 5–8, 2022, Proceedings*. Springer, Springer.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Juliano Rabelo, Randy Goebel, mi-young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2022. Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. *The Review of Socionetwork Strategies*.

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.

Mina Rezaei, Farzin Soleymani, Bernd Bischl, and Shekoofeh Azizi. 2021. Deep bregman divergence for contrastive learning of visual representations. *arXiv preprint arXiv:2109.07455*.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient Few-Shot Learning Without Prompts. In *36th Conference on Neural Information Processing Systems*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.

A   Hyper-parameter Optimization

Continued Pre-training: We define the search space based on previous studies such as Rezaei et al. (2021) and Gao et al. (2021). For the contrastive Bregman divergence, we benchmark the performance for the first-stage hyper-parameters on the downstream task to tune the respective hyper-parameters. We use mean pooling for all settings. The learning rate, the total optimization steps, the use of a batch-norm layer, the σ parameter, the number of sub-networks g, and the batch size are grid-searched. Temperature (.1) and the input length to 4096 are fixed beforehand. The learning rate for these models was 3e-5. We run 50.000 optimization steps for each model.

Training for classification tasks: We used AdamW as an optimizer. Bayesian optimization is used to tune the hype-rparameters learning rate, number of epochs and batch size. We use mean pooling for all settings. Early stopping is set to a patience score of 3.[3] These parameters were fixed after some early experiments. We use a learning rate of 1e-4 and run ECTHR and SCOTUS for 20 and

---

[3] We also experimented with other patience scores but experiments suggest that 3 epochs results in the best performance.

Hyper-parameters	μ-F_1	m-F_1	μ-F_1	m-F_1	μ-F_1	m-F_1	μ-F_1	m-F_1	μ-F_1	m-F_1
$g \in [2,5,8,10,20]$	74.1	<u>64.3</u>	<u>74.3</u>	62.2	72.1	61.0	**75.2**	**67.7**	73.9	63.1
$\sigma \in [1,1.5,2,2.5,3]$	73.3	63.0	73.6	61.2	**75.2**	**67.7**	73.0	62.0	<u>73.9</u>	<u>64.1</u>
Steps $\in [10\text{-}50k]$	74.21	62.79	74.14	63.44	**75.6**	<u>63.5</u>	73.5	63.36	<u>75.2</u>	**67.7**
Batch size $\in [2,4,8,12]$	**75.2**	**67.7**	<u>74.36</u>	<u>64.2</u>	73.9	62.6	74.21	62.9	-	-
$\lambda \in [.1,2,4,5,10]$	<u>75.1</u>	<u>65.3</u>	**75.2**	**67.7**	74.79	63.4	74.1	63.7	**75.2**	64.0

Table 4: m-F_1 & μ-F_1 performance benchmark for end-to-end training with SCOTUS

30 epochs respectively for the MLP head setting. For MIMIC we used 10 epochs for the MLP head and had to truncate the maximum sequence length to 2048 due to computational constraints. For each task we compared multiple different training checkpoints of our encoder. The reported results are the best performing checkpoints.

B Number of parameters

Table 5 shows the number of parameters for the different models. Modding the transformer to a Longformer adds 6M parameters for LegalBERT small and 24M parameters for BioBERT medium. By working with LegalBERT-small and BioBERT-base we cover both small and medium sized models.

Model	#Params
BioBert$_{Base}$	109M
Longformer$_{Base}$	148M
LegalBERT$_{small}$	35M
Longformer$_{Legal\text{-}DA + SimCSE + Bregman}$	41M
Longformer$_{Bio\text{-}DA}$	134M
Longformer$_{MLP}$	.27M

Table 5: Number of Parameters for the Longformer variants.

C Pooling methods

We evaluate Mean, Max and [CLS] pooling. Results for end-to-end fine-tuning can be found in the table 6. Our results show that using mean pooling during continued pre-training in combination with max-pooling for classification could further enhance the performance instead of using the same pooling method for both stages.

D Neural network Architecture

Our model contains two linear layers with one activation layer and two batch normalization layers.

Pooling operator	m-F_1	μ-F_1
Mean + Max Pooling	**78.3**	**70.6**
Mean Poolig	76.9	68.1
Max Pooling	<u>77.6</u>	<u>69.5</u>
[CLS] Pooling	77.1	69.5

Table 6: Test results for various pooling operators with end-to-end tuning on SCOTUS for Longformer$_{DA}$.

We also compare the model without batch normalization layers. The comparison is made on the SCOTUS dataset using end-to-end fine-tuning. One can see that removing batch normalization worsens performance.

Normalization	m-F_1	μ-F_1
Batch Norm	**75.6**	**63.5**
w/o Batch Norm	72.5	63.1

Table 7: F1 performance for ablation model without batch norm layers for end-to-end fine-tuning on SCOTUS.

A For every submission:

☑ A1. Did you describe the limitations of your work?
Unnumbered

☒ A2. Did you discuss any potential risks of your work?
We don't see any direct potential risk.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
Section 2

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We re-use already available public resources.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A.2

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Because we don't.

D ☒ **Did you use human annotators (e.g., crowdworkers) or research with human participants?**

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.