Towards Meaningful Paragraph Embeddings for Data-Scarce Domains: a Case Study in the Legal Domain

Elize Herrewijnen[1], M.Sc. (PhD Candidate), Dennis F W Craandijk[1], M.Sc.

[1]*National Police Lab AI, Utrecht University, Heidelberglaan 8, 3584 CS Utrecht, The Netherlands*

Abstract

Creating meaningful text embeddings using BERT-based language models involves pre-training on large amounts of data. For domain-specific use cases where data is scarce (e.g., the law enforcement domain) it might not be feasible to pre-train a whole new language model. In this paper, we examine how extending BERT-based tokenizers and further pre-training BERT-based models can benefit downstream classification tasks. As a proxy for domain-specific data, we use the European Convention of Human Rights (ECtHR) dataset. We find that for down-stream tasks, further pre-training a language model on a small domain dataset can rival models that are completely retrained on large domain datasets. This indicates that completely retraining a language model may not be necessary to improve down-stream task performance. Instead, small adaptions to existing state-of-the-art language models like BERT may suffice.

Keywords

Transformers, BERT, Language Models, Legal Text Classification, ECtHR dataset, Text Embeddings

1. Introduction

Large language models like *BERT* have proven their use in natural language processing (NLP) [1, 2, 3]. By pre-training the language model on a large amount of textual data, it learns to represent text in a semantically meaningful way. This representation is also called an embedding. Such embeddings can be learned without supervision and can effectively capture relevant information for downstream tasks like question answering and classification.

Various work has shown that tailoring language models to specific domains is beneficial for downstream task solving [4], for example in the financial [5] and legal [6] domain. In the law enforcement domain, language models may be used to effectively process large amounts of text data (e.g. police reports) [7, 8]. Applying generic language models to encode such data may result in suboptimal embeddings, as the model has not learned to encode domain-specific features. Pre-training language models from scratch requires a large amount of data and compute, which might not be available in domains like law enforcement. In this work, we create a domain-specific language model without requiring large amounts of training data. We use a well-known dataset from the legal domain (ECtHR). We make our code available on GitHub.[1]

2. Related work

2.1. Masked language modelling (MLM)

Large language models like *BERT* [9] are trained with the masked language modelling (MLM) objective. The model learns to predict a masked token based on the surrounding tokens, which allows it to generate a meaningful language representation. The representations can be used to train ML models for downstream tasks like sentiment classification or question answering, without the need for hand-crafted feature engineering [9]. A single text embedding can be reused to train various downstream task models, without requiring a task-specific architecture.

2.2. Domain-adapted tokenizers

Nayak et al. [10] find that the *BERT* tokenizer inadequately handles misspellings and Out-of-Vocabulary (OOV) words, which negatively impacts the efficiency and semantic meaning of the embeddings. Benamar et al. [11] show that adding new words to a model's vocabulary is easier than improving the representation of words that are already present.

2.3. Further pre-training language models

Since the introduction of *BERT*, many domain-specific language models have been put on the market, for example in the clinical [12], financial [13], biomedical [14], and legal [6, 15] domain. Using embeddings from domain-specific language models has a positive effect on the performance of various downstream-task NLP models, be-

Proceedings of the Sixth Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2023), June 23, 2023, Braga, Portugal.

*Corresponding author.

e.herrewijnen@uu.nl (E. Herrewijnen); d.f.w.craandijk@uu.nl (D. F. W. Craandijk)

0000-0002-2729-6599 (E. Herrewijnen); 0000-0001-6815-7053 (D. F. W. Craandijk)

[1]https://github.com/UtrechtUniversity/Meaningful-Paragraph-Embeddings-for-Data-Scarce-Domains

cause the text embeddings contain more domain-specific information.

For the legal domain, Limsopatham [16] compare the newly pre-trained models by Chalkidis et al. [6] and Zheng et al. [15] and find that both legal domain-specific models outperform generic language models like *BERT*. However, these models inadequately encode long legal texts, as parts of the inputs are truncated to fit into the language model.

In the clinical domain, Lamproudis et al. [17] show that further pre-trained BERT models on in-domain data outperform generic BERT models, after a single training epoch. In this paper, we investigate whether this also applies to the ECtHR dataset, which is representative for the legal domain.

3. Methods

Creating meaningful text embeddings requires multiple steps: first, a **tokenizer** model tokenizes the text. This tokenization is used by an **encoder** model to create an embedding. Finally, this embedding can be used by a **predictor** model to perform a downstream task. We now describe how the tokenizer, language model, and predictor can be modified to achieve meaningful embeddings in scarce-data domains.

3.1. Dataset

The European Court of Human Rights (ECtHR) handles alleged violations of European Convention of Human Rights (ECHR) articles.[2] We use this dataset as a proxy for law enforcement datasets, as these datasets often consist of long texts with domain-jargon in our experience. The ECtHR dataset as introduced by Chalkidis et al. [18] contains 11k legal cases, containing facts (a list of paragraphs representing the facts of the case such as events), allegedly violated articles, violated articles, and silver allegation rationales (relevant facts identified using a regular expression) and gold allegation rationales (relevant facts annotated by a legal expert).

To further pre-train our language model, we use all facts in training split as used by Chalkidis et al. [18], further split into a total of 588090 sentences. For our down-stream task, we use the *violated articles* as labels, resulting in a multi-label classification task. Due to the class imbalance in the dataset, we only retain the 10 most common classes (see Table 1), and adopt the same train, dev, and test splits as Chalkidis et al. [18] for training the classification model. As shown in Table 1, article types vary in number of facts and number of characters, which we statistically tested as significant using a Two-Sample t-Test.

Art.	Name	Supp.	Facts	Char.
6	Right to a fair trial	4704	19	6057
P1-1	Protection of property	1421	16	5690
5	Right to liberty and security	1368	37	15036
3	Prohibition of torture	1349	42	18569
13	Right to an effective remedy	1238	33	13118
8	Right to respect for private and family life	710	31	14755
2	Right to life	505	59	26102
10	Freedom of expression	291	19	12371
14	Prohibition of discrimination	141	25	14014
11	Freedom of assembly and association	110	24	13143
(Other articles)		896	24	13518

Table 1
Retained **art**icles and their **supp**ort, average number of **facts**, and average number of **char**acters per document in the training split.

3.2. Language models

As baselines for our analysis, we select four *BERT*-based language models that have shown their applicability to NLP in the legal domain.

BERT-ML The BERT base multilingual cased (*BERT-ML*) [9] is a multi-language model pre-trained on the top 104 languages with the largest Wikipedia corpus. It is a powerful model for capturing generic text data, and can effectively be fine-tuned for downstream tasks [19].

LEGAL-BERT The *LEGAL-BERT* model is trained from scratch using the same approach as *BERT*, but on 12 GB English legal texts (e.g., legislation, court cases, contracts) from publicly available sources [6]. This model outperforms the *BERT* model when fine-tuned for legal classification tasks [16].

RoBERTa The *RoBERTa* model by Liu et al. [20] is a version of *BERT*, that is trained on a much larger (x10) English language corpus using a dynamic masking technique. This allows the model to produce more robust and generalizable embeddings, outperforming *BERT* on various NLP tasks [20].

Longformer The *Longformer* model by Beltagy et al. [21] builds on *RoBERTa*, but expands the max input length to 4096 tokens. The model is further pre-trained on large generic texts like news and web pages, and outperforms

[2]See https://www.echr.coe.int/Documents/Convention_ENG.pdf for an extensive description of the convention.

Longformer^f	appeal, applicant, applicants, april, august, december, decision, detention, district, february, further, hearing, investigation, january, judgment, july, june, march, november, october, proceedings, prosecutor, regional, september, submitted
BERT-ML^f	applicant, applicants, detention, january, june, mr, october, prosecutor

Table 2
Domain-specific words newly added to the tokenizers.

RoBERTa on long document NLP tasks [21]. Note that the increased max input length renders the model more resource-expensive.

3.3. Tokenizer

Effective text embeddings begin with the tokenization of the input text. A tokenizer tokenizes a text using a pre-defined vocabulary. If a word is not in the vocabulary, it is distributed across vocabulary tokens (e.g., *applicant* becomes *app*, *lica*, and *nt*). Due to their architecture, encoder models limit the max input length (usually 512 tokens). The tokenizer model should respect this limit, which usually results in input truncation. However, truncation may negatively affect downstream task performance [6] as information is lost. Thus, a larger vocabulary reduces the number of tokens required to tokenize a text, allowing more information to be captured. While a large vocabulary might seem desirable, it also increases the number of parameters the encoder model has to learn, negatively affecting training time and memory requirements. Hence, a tokenizer should be able to capture as much relevant information as possible while keeping the number of parameters (i.e., the vocabulary) manageable.

While a tokenizer that is specifically trained on domain data may be able to capture texts most effectively, it may be unfeasible to train a new tokenizer; even when training data are available, the encoder model also needs to be retrained, which is a resource- and time-consuming task. Therefore, *extending* a tokenizer with domain-specific tokens may be more feasible. By adding domain-specific words, these words are not split up during tokenization, which leaves more space for other tokens. Moreover, the encoder model might be able to capture information concerning the domain-specific tokens, allowing more meaningful embeddings. For example, the *LEGAL-BERT* model (which contains domain-specific tokens) only requires a single token for the word '*applicants*', while the *BERT-ML* tokenizer requires the tokens *'app'*, *'lica'*, and *'nts'*.

We select the top 1% most common words in the dataset based on relative frequency using the Scikit-learn [22] CountVectorizer, and add only the yet unknown tokens to the *BERT-ML* and *RoBERTa* tokenizers. As shown in Table 2, novel words are related to the legal domain, for example 'applicant', 'prosecutor', 'detention' and month names. In total, 25 and 9 new words are added to the tokenizer vocabularies, respectively.

3.4. Encoder models

We use the extended tokenizers to further pre-train two encoder models on the ECtHR training set on a machine with 2 50 GB NVIDIA RTX A6000 cards:[3] using the script provided by Devlin et al. [9], we further pre-train the *BERT-ML* model for 1 epoch with a batch size of 16, which takes approximately 40 minutes. Using the script provided by Beltagy et al. [21], we convert a *RoBERTa* model to a *Longformer* model, and further pre-train the model for 3000 steps with a batch size of 24, which takes approximately 2 days. We will further refer to these further pre-trained encoder models as *BERT-ML^f* and *Longformer^f*.

3.5. Classification model

We employ a convolutional neural network to classify the documents: for every fact in the document, an embedding is retrieved using one of the models from 3.2; then, the list of embeddings is stacked and fed to the network. The network consists of 3 1-dimensional convolutional layers (768 × 768, kernel-size 1), followed by 3 linear layers (768 × 10). Finally, the mean of predictions for all facts is taken to compute the final prediction. A benefit of this stacked approach is that every fact receives an embedding, retaining more information than creating a single embedding for the whole document by concatenating facts. The model is trained using weighted *BCE Loss* and the *Adam* optimizer, for 15 epochs (no early stopping) on a machine with 2 25 GB NVIDIA GeForce RTX 3090 cards.[4] Note that the parameters of the encoder model as described in the previous subsection remain frozen. Furthermore, the focus of this paper lies on finding the meaningful embeddings, and not on the classification accuracy of the classification model: we investigate how well the different embeddings allow the classification model to learn the task.

4. Results

In the following section, we discuss our results for both tokenization and classification.

[3]Note that the training set is only 85 Mb.
[4]More model training details can be found on the Github page.

	BERT-ML	LEGAL-BERT	BERT-ML^f	RoBERTa	Longformer	Longformer^f
I	512	512	512	512	4096	4096
V	119547	**30522**	119556	50265	50290	50265
TD	2248	**2048**	2183	2129	2193	2129
UT	36064	**23726**	36065	36981	36971	36981
mDT ↓	107	92	105	95	**2**	**2**
tDT ↓	967707	831087	947454	857869	25074	**24127**

Table 3
Statistics of tokenization as performed by various tokenizer models. Abbreviations are as follows: I: max input length, **V**: vocabulary size, **TD**: mean number of tokens per document, **UT**: number of unique tokens in dataset, **mDT**: mean number of tokens discarded per document, **tDT**: total number of tokens discarded in dataset.

Article	BERT-ML	LEGAL-BERT	BERT-ML^f	RoBERTa	Longformer	Longformer^f	support
6	.53	.50	**.55**	.50	.49	.50	299
P1-1	**.53**	.05	.02	.03	.39	.03	122
5	.01	.36	**.39**	.01	.14	.27	166
3	.22	.43	**.47**	.15	.22	.43	189
13	.24	.25	**.28**	.20	.23	.07	79
8	.02	.0	0	.0	.0	.0	123
2	.19	.48	**.49**	.38	.43	.32	56
10	.12	**.17**	.08	.05	.0	.12	77
14	.0	.0	.0	.0	.0	.0	16
11	.0	.0	.0	.0	.0	.0	37
Other							155

Table 4
F1-scores for the classification model, trained on embeddings from various encoder models on the test set.

4.1. Tokenization

We compare the tokenization result of the tokenizer models introduced in Section 3.2, by tokenizing the complete ECtHR dataset. Specifically, we note the following:

- The mean number of tokens required to tokenize a document (TD);
- The total number of unique tokens in all documents as tokenized by the tokenizer (UT);
- The mean number of tokens discarded for a document due to truncation (mDT);
- The sum of discarded tokens in all documents (tDT).

For all of the above holds that the lower the values, the more efficient the tokenizer is. The results reported in Table 3 show that the *LEGAL-BERT* tokenizer is most efficient in tokenizing input texts. The tokenizer requires the fewest tokens to tokenize documents, discards the fewest tokens in comparison to other 512-limited tokenizers, while also having the smallest vocabulary. The *Longformer* models discard the fewest tokens overall, but require more tokens than the *LEGAL-BERT* tokenizer. Extending existing tokenizers slightly decreases the number of discarded tokens (average of 2 for both tokenizers). Thus, retraining the tokenizer model decreases the amount of removed information, but may still be insufficient for long documents.

4.2. Classification

As the classification task is an unbalanced multi-label problem, we note the F1-scores in Table 4. We focus on the classification model's ability to identify independent classes, instead of the average F1-score. If the classification model is unable to identify a class (i.e., $F1 = 0$), we take this as an indication that the embedding does not contain relevant information about that class. Related work has noted that the multi-label classification is difficult to solve [18]. Our classification performance is also fair, but a clear difference between embeddings is visible:

- *BERT-ML^f* embeddings outperform *BERT-ML* embeddings on most classes, indicating that extending existing tokenizers and further pre-training existing language models may be sufficient for solving domain-specific use-cases.
- *BERT-ML* embeddings generally capture sufficient information for the classification task, which is in line with work on domain adaptation of language models in the clinical domain [23].
- *LEGAL-BERT* embeddings generally perform well, but are closely rivalled by the *BERT-ML^f* and *Longformer^f* embeddings, showing the potential of using the combination of further pre-training existing language models.
- The *Longformer* embeddings outperform *RoBERTa* embeddings, but not *BERT*-based embeddings, showing that increasing the max input length may not be necessary for some tasks.

5. Limitations and future work

This work mainly focuses on the effect of further pre-training *BERT*-based language models on limited domain-specific data. As we do not investigate or optimize the pre-training procedure of our BERT models, a highly relevant point for future work is investigating how BERT models can be (more) effectively (further) pre-trained on (scarce) domain-specific data. Furthermore, we used a multilingual BERT model as a starting point, which may negatively affect performance on down-stream tasks.

Another limitation is that the performance of the classification model (Section 4.2) is rather low, which is due to the minimal effort put into the model. Related work (e.g., Chalkidis et al. [18]) show much higher F1-scores using more advanced (and tested) classification models. Moreover, a more throughout error analyses might give insight in the documents that are typically miss-classified by the classification model, and how pre-training the encoder models impacts classification behaviour.

A point of caution is that pre-training a language model like BERT on domain data may introduce domain-specific bias, especially when the domain dataset misrepresents identity groups (e.g., males are over-represented) [24]. To apply language models like BERT in the law enforcement domain, the possibility of introduced bias should be investigated in future work.

Finally, a limitation is the generalizability of the dataset and tasks; this work only looks at the effect of pre-training on one well-known domain-specific dataset (ECtHR), task (violated article classification). We expect that our findings generalize across other domain-specific datasets and tasks, especially for long texts with domain-jargon. Nevertheless, future work is required to further validate this expectation.

6. Conclusion & discussion

In this paper, we investigate the effect of further pre-training large language models on domain-specific data. In order to test this on scarce-domain data, we use the ECtHR dataset as a surrogate (Section 3.1), and further pre-train a *BERT-ML* and a *Longformer* language model on this data.

We find that extending tokenizers with domain-specific tokens reduces the number of tokens discarded, albeit slightly (Section 4.1). Retraining a tokenizer results in a much more efficient tokenization result, but also requires more data and retraining an encoder model from scratch, which might be unfeasible. In a data-scarce or resource-scare setting, extending the tokenizer may be a good alternative, as fewer data is required to further pre-train the encoder model.

Embeddings constructed by the original *BERT-ML* adequately encode legal domain-specific information, but a completely retrained language model may be beneficial for some classification problems (Section 4.2). Moreover, in scarce-data settings, further pre-training *BERT*-based models using small amounts may be a feasible alternative to training a language model from scratch. In particular, the combination of adding domain-specific tokens to the tokenizer and further pre-training the language model on a small dataset is a promising direction for future research. Whether our findings generalize across other domains and tasks is a question for future work.

References

[1] M. V. Koroteev, Bert: a Review of Applications in Natural Language Processing and Understanding, arXiv preprint arXiv:2103.11943 (2021).

[2] S. Aftan, H. Shah, A Survey on Bert and Its Applications, in: 2023 20th Learning and Technology Conference (L&T), IEEE, 2023, pp. 161–166.

[3] P. Xia, S. Wu, B. Van Durme, Which BERT? A Survey Organizing Contextualized Encoders, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 7516–7533.

[4] X. Ma, P. Xu, Z. Wang, R. Nallapati, B. Xiang, Domain Adaptation with Bert-based Domain Classification and Data Selection, in: Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019), 2019, pp. 76–83.

[5] B. Peng, E. Chersoni, Y.-Y. Hsu, C.-R. Huang, Is Domain Adaptation Worth your Investment? Comparing Bert and Finbert on Financial Tasks, in: Proceedings of the Third Workshop on Economics and Natural Language Processing, 2021, pp. 37–44.

[6] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, LEGAL-BERT: The Muppets straight out of Law School, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 2898–2904.

[7] V. Saxena, N. Rethmeier, G. Van Dijck, G. Spanakis, VendorLink: An NLP approach for Identifying & Linking Vendor Migrants & Potential Aliases on Darknet Markets, arXiv preprint arXiv:2305.02763 (2023).

[8] B. W. Hung, S. R. Muramudalige, A. P. Jayasumana, J. Klausen, R. Libretti, E. Moloney, P. Renugopalakrishnan, Recognizing Radicalization Indicators in Text Documents using Human-in-the-Loop Information Extraction and NLP Techniques, in: 2019 ieee international symposium on technologies for homeland security (hst), IEEE, 2019, pp. 1–7.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), 2019, pp. 4171–4186.

[10] A. Nayak, H. Timmapathini, K. Ponnalagu, V. G. Venkoparao, Domain Adaptation Challenges of

Bert in Tokenization and Sub-word Representations of Out-of-vocabulary Words, in: Proceedings of the First Workshop on Insights from Negative Results in NLP, 2020, pp. 1–5.

[11] A. Benamar, C. Grouin, M. Bothua, A. Vilnat, Evaluating Tokenizers Impact on Oovs Representation with Transformers Models, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, 2022, pp. 4193–4204.

[12] M. Sushil, S. Suster, W. Daelemans, Are We There Yet? Exploring Clinical Domain Knowledge of Bert Models, in: Proceedings of the 20th Workshop on Biomedical Language Processing, 2021, pp. 41–53.

[13] D. Araci, FinBERT: Financial Sentiment Analysis with Pre-trained Language Models, arXiv preprint arXiv:1908.10063 (2019).

[14] W. Tai, H. T. Kung, X. Dong, M. Comiter, C.-F. Kuo, exBERT: Extending Pre-trained Models with Domain-specific Vocabulary Under Constrained Training Resources, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 1433–1439.

[15] L. Zheng, N. Guha, B. R. Anderson, P. Henderson, D. E. Ho, When does Pretraining Help? Assessing Self-supervised Learning for Law and the Casehold dataset of 53,000+ Legal Holdings, in: Proceedings of the eighteenth international conference on artificial intelligence and law, 2021, pp. 159–168.

[16] N. Limsopatham, Effectively Leveraging Bert for Legal Document Classification, in: Proceedings of the Natural Legal Language Processing Workshop 2021, 2021, pp. 210–216.

[17] A. Lamproudis, A. Henriksson, H. Dalianis, Evaluating Pretraining Strategies for Clinical Bert Models, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, 2022, pp. 410–416.

[18] I. Chalkidis, M. Fergadiotis, D. Tsarapatsanis, N. Aletras, I. Androutsopoulos, P. Malakasiotis, 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, Association for Computational Linguistics, Online, 2021, pp. 226–241. doi:10.18653/v1/2021.naacl-main.22.

[19] C. Sun, X. Qiu, Y. Xu, X. Huang, How to Fine-tune Bert for Text Classification?, in: Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18, Springer, 2019, pp. 194–206.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A Robustly Optimized Bert Pretraining Approach, arXiv preprint arXiv:1907.11692 (2019).

[21] I. Beltagy, Matthew E. Peters, Arman Cohan, Longformer: The Long-document Transformer, arXiv:2004.05150 (2020).

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[23] C. Lin, S. Bethard, D. Dligach, F. Sadeque, G. Savova, T. A. Miller, Does Bert need Domain Adaptation for Clinical Negation Detection?, Journal of the American Medical Informatics Association 27 (2020) 584–591.

[24] F. Elsafoury, S. Katsigiannis, N. Ramzan, On Bias and Fairness in NLP: How to have a fairer text classification?, arXiv preprint arXiv:2305.12829 (2023).