Model Documentation Form

This Form includes all the information to be documented as part of Measure 1.1 of the Transparency Chapter of the Code of Practice. Crosses on the right indicate whether the information documented is intended for the AI Office (AIO), national competent authorities (NCAs) or downstream providers (DPs), namely providers of AI systems who intend to integrate the general-purpose AI model into their AI systems. Whilst information intended for DPs should be made available to them proactively, information intended for the AIO or NCAs is only to be made available following a request from the AIO, either ex officio or based on a request to the AIO from NCAs. Such requests will state the legal basis and purpose of the request and will concern only items from the Form strictly necessary for the AIO to fulfil its tasks under the AI Act at the time of the request, or for NCAs to exercise their supervisory tasks under the AI Act at the time of the request, in particular to assess compliance of high-risk AI systems built on general-purpose AI models where the provider of the system is different from the provider of the model.

Any elements of information from the Model Documentation Form shared with the AIO and NCAs shall be treated in accordance with the confidentiality obligations and trade secret protections set out in Article 78 AI Act.

Date this document was last updated: 01/09/2026

Document version number: Click or tap here to enter text.

General information		AIO	NCAs	DPs
Legal name for the model provider:	Nasjonalbiblioteket	☒	☒	☒
Model name:	borealis-embed-212M	☒	☒	☒
Model authenticity:	We provide a package that proves that Nasjonalbiblioteket has signed the files and that Hellenic Academic and Research Institutions CA have approved that we are indeed Nasjonalbiblioteket. Verification can be done following https://ai.nb.no/verify. 6c2cad752fb5bd554e82a92034c24df54c03b149787b5645044ff0ac3b27c008 model.safetensors	☒	☒	☐
Release date:	01/09/2026 Date when the model was first released through any distribution channel.	☒	☒	☒
Union market release date:	01/09/2026 Date when the model was placed on the Union market.	☒	☒	☒
Model dependencies:	N/A	☒	☒	☒

Model properties						AIO	NCAs	DPs
Model architecture:	The model is based on the Gemma3Text architecture with a few changes and fine-tuned as a sentence transformer with mean pooling.					☒	☒	☒
Design specifications of the model:	For the architecture of the model, we decided to use Gemma3Text for easy adaptation on Huggingface and because the architecture is well tested and leads to well performing models. For easier adaptation to an embedding model, we decided to include masked language modelling in our training (in the form of masked next token prediction). We chose a size that corresponds to most base named models for speed and low-memory usage. This model is for creating sentence/passage embeddings to be used in embedding-based tasks like bitext mining, classification, clustering, and retrieval.					☒	☒	☐
Input modalities:	☒Text	☐Images	☐Audio	☐Video	If any other please specify:	☒	☒	☒
For each selected modality please include maximum input size or write 'N/A' if not defined	Maximum size: 8192 tokens	Maximum size: …	Maximum size: …	Maximum size: …	Maximum size: …	☒	☐	☒
Output modalities:	☐Text	☐Images	☐Audio	☐Video	768 dimension vector	☒	☒	☒
For each selected modality please include maximum output size or write 'N/A' if not defined	Maximum size: …	Maximum size: …	Maximum size: …	Maximum size: …	Maximum size: 1 vector	☐	☐	☒
Total model size:	$2.1*10^8$					☒	☐	☐
The range within which the total number of parameters falls.	☒1—500M	☐500M—5B	☐ 5B—15B	☐15B—50B		☐	☒	☒
	☐50B—100B	☐100B—500B	☐500B—1T	☐>1T				

Methods of distribution and licenses		AIO	NCAs	DPs
Distribution channels:	The model is available on HuggingFace at https://huggingface.co/NbAiLab/borealis-embed-212m	☒	☒	☐

	The model is available on HuggingFace at https://huggingface.co/NbAiLab/borealis-embed-212m	☐	☐	☒
License:	NB-license (https://huggingface.co/NbAiLab/borealis-embed-212m/blob/main/LICENSE)	☒	☒	☐
	NB-license (https://huggingface.co/NbAiLab/borealis-embed-212m/blob/main/LICENSE)	☐	☐	☒
	NB-license (https://huggingface.co/NbAiLab/borealis-embed-212m/blob/main/LICENSE)	☒	☐	☒
Use		**AIO**	**NCAs**	**DPs**
Acceptable Use Policy:	None	☒	☒	☒
Intended uses:	Sentence embeddings and retrieval	☒	☒	☒
Type and nature of AI systems in which the general-purpose AI model can be integrated:	Any system that would need a vector representation of text.	☒	☒	☒
Technical means for model integration:	The model is available on HuggingFace. See the model card for example usage with SentenceTransformers and transformer libraries.	☐	☐	☒
Required hardware:	Any GPU with at least 4GB of VRAM or any CPU with at least 4GB of RAM.	☐	☐	☒
Required software:	Transformers: https://github.com/huggingface/transformers@v4.49.0-Gemma-3	☐	☐	☒
Training process		**AIO**	**NCAs**	**DPs**
Design specifications of the training process:	The pretraining was broken into three separate stages. In all stages, we used a mixture of three training objectives; Causal Language Modelling (CLM), Diffusion Language Modelling (DLM), and Masked Next Token Prediction (MNTP) at the rates of 75%, 18.75%, and 6.25% respectively. To optimize our models, we use gradient-based optimization via the NorMuon optimizer with polar express coordinates for the Newton-Shultz parameters. In Stage 1, we start with randomly initialized weights and train for 71,526 steps with 2,500 steps of warm-up to reach a maximum learning rate of $7.5*10^{-4}$. This represents approximately 300B tokens and 6 epochs of our pretraining data. We train with a batch size of 4M tokens and a max sequence length of 2048. In Stage 2, we train for a further 150B tokens at a constant learning rate with a longer max sequence length of 8192. Finally, in Stage 3 we decay the learning rate to 0 for 150B tokens using a 1 – sqrt schedule. In total, we pretrain approximately 12 epochs of our data. The embedding training process employs a contrastive learning methodology to fine-tune the resulting encoder, optimizing it to map semantically related text into a shared vector space. By utilizing MultipleNegativesRankingLoss (InfoNCE), the model learns to minimize the distance between anchor-positive pairs while maximizing the distance to contradiction-based negatives. A key design choice is the use of in-batch negatives, which treats all other sentences within a training batch as additional negative examples to enhance the efficiency of the embedding signal. Additionaly, we experiemented with the directions for the datasets and scale values to identify configurations that work well. To ensure training stability and performance, the configuration incorporates warmup steps to prevent aggressive early weight updates and uses gradient accumulation to simulate larger batch sizes for a stronger contrastive signal. Finally, the BatchSamplers.NO_DUPLICATES parameter is critical for preventing training noise by ensuring that no identical sentences appear within the same batch.	☒	☒	☐
Decision rationale:	For selecting hyperparameters, we did a quick scaling law experiment and extrapolated the predicted optimal learning rates. To make our training more efficient and add the possibility of repeating the data more often, we trained with multiple objectives within each batch (Causal Language Modelling, Diffusion Language Modelling, and Masked Next Token Prediction) at a rate of 75% CLM, 18.75% DLM, and 6.25% MNTP. Using multiple objectives is based on the paper: "DUAL-OBJECTIVE LANGUAGE MODELS: TRAINING EFFICIENCY WITHOUT OVERFITTING". We also do half the training with a lower max sequence length to accelerate our training. Finally, we use NorMuon to make our training more efficient as it lets our model reach a lower validation loss faster as compared to AdamW. Decisions made during the sentence embedding training process were grounded in a combination of best practices and empirical evidence. The embedding model architecture is the standard recommended by SentenceTransformers. Training hyperparameters and datasets were selected through training sweeps and model evaluations to estimate which configuration produces the best results.	☒	☒	☐

Information on the data used for training, testing, and validation						AIO	NCAs	DPs
Data type/modality: *Select all that apply.*	☒Text	☐Images	☐Audio	☐Video	If any other please specify:	☒	☒	☒
Data provenance: *Select all that apply* For definitions of each listed category, see the Template for the Public Summary of the Training Content of General-Purpose AI models provided by the AI Office	☒Web crawling	☒Private non-publicly available datasets obtained from third parties	☐User data			☒	☒	☒
	☒Publicly available datasets		☐Data collected through other means					
	☐Synthetic data that is not publicly accessible (when created directly by or on behalf of the provider)		If any other please specify:					

How data was obtained and selected:	Pretraining data was obtained via a combination of downloading publicly available datasets from the internet (web crawls), using our own publicly available data (books, manuscripts, crawls), and using proprietary data from publishers (newspapers). The NbAiLab/mnli-norwegian dataset was created by machine-translating the original English Multi-Genre Natural Language Inference (MultiNLI) corpus into Norwegian Bokmål using Google Translate. We transform the MNLI dataset into a triplet format suitable for contrastive learning. We generate training samples by selecting an anchor sentence and pairing it with a randomly sampled positive (entailment) and negative (contradiction) to create the final triplet structure. Additionally, we created an internal dataset using Norwegian newspapers. Each instance in the data set is pair mapping the title and lead text from the article to the body of the article. The data was sourced from newspapers provided by Schibsted and Klassekampen from 1995 to 2024. Remaining datasets were chosen from a sample of existing datasets using forward stepwise selection. Starting with the NLI dataset, we trained small models, adding one new dataset at a time to the previous set of datasets. After evaluation, the dataset that offered the most improvement to model performance was added to the set of datasets for the next iteration. We removed datasets that negatively impacted performance during this process. In addition to NbAiLab/mnli-norwegian and our internal newspaper dataset, the resulting dataset contains the Norwegian subsets of sentence-transformers/parallel-sentences-opus-100, a subset of the retrieval instances from DDSC/nordic-embedding-training-data, and Norwegian Wikipedia articles from lightonai/embeddings-pre-training.	☒	☒	☐
Number of data points:	Our pretraining dataset consists of 51,287,756,354 tokens (using our own tokenizer) from 66,512,166 documents for the training set. For the validation set, we use 16,777,216 tokens (split into 8,192 sequences of length 2,048 or 2,048 sequences of length 8,192). Our embedding dataset consists of 3,480,841 training samples.	☐	☒	☐
	Our pretraining dataset consists of 51,287,756,354 tokens (using our own tokenizer) from 66,512,166 documents for the training set. For the validation set, we use 16,777,216 tokens (split into 8,192 sequences of length 2,048 or 2,048 sequences of length 8,192). Our embedding dataset consists of 3,480,841 training samples.	☒	☐	☐
Scope and main characteristics:	The aurora-open pretraining dataset contains a varied set of sources, including data from the internet, government documents, code, math, books, and more. The highest prioritized languages are Norwegian Bokmål and Nynorsk, as well as the Sami languages, followed by other Nordic languages and English. The original MNLI dataset covers 10 domains. The thematic and geographic context of our subset reflects this original source but are translated to Norwegian to provide better Norwegian performance. The newspaper pairs cover regional and global news from a variety of Norwegian newspapers. sentence-transformers/parallel-sentences-opus-100 contains English-Norwegian sentence pairs drawn from diverse web, subtitle, and document sources. Retrieval instances from DDSC/nordic-embedding-training-data are synthetically generated, open-domain triplets in Danish, Norwegian, and Swedish. Norwegian Wikipedia articles from lightonai/embeddings-pre-training are general-knowledge documents covering both global and regional topics.	☒	☒	☐
Data curation methodologies:	The aurora-open pretraining dataset was built by collecting a variety of openly available datasets from the internet, as well as using the National Library's own open collection to produce text data. Data was cleaned and filtered using a script largely based on the FineWeb2 filtering script. Additionally, we have a filter to comply with robots.txt for internet data, a filter to remove blacklisted domains according to WIPO ALERT, and a filter to remove newspaper domains. The NbAiLab/mnli-norwegian dataset provides information about translation methods. Further processing includes generating training samples by selecting an anchor sentence and pairing it with a randomly sampled positive (entailment) and negative (contradiction) to create the final triplet structure.	☒	☒	☒

	Additionally, we created an internal dataset using Norwegian newspapers. Each instance in the data set is pair mapping the title and lead text from the article to the body of the article. Articles were filtered following the number of paragraphs available; those with fewer than the 25th percentile or more than the 99th percentile were removed. The data was sourced from newspapers provided by Schibsted and Klassekampen from 1995 to 2024. The remaining datasets were chosen from openly available embedding datasets through an iterative process of selection according to performance gains.			
Measures to detect unsuitability of data sources:	For the aurora-open pretraining dataset, filters were added to comply with robots.txt, avoid blacklisted domains according to WIPO ALERT, and avoid newspaper domains. No measures were taken to detect unsuitable sources for the embedding dataset.	☒	☒	☐
Measures to detect identifiable biases:	No measures were taken to detect identifiable biases in the aurora-open pretraining dataset. No measures were taken to detect identifiable biases in the embedding dataset. However, when choosing sources for the articles, we tried to include articles from a variety of newspapers and locations.	☒	☒	☐

Computational resources (during training)		**AIO**	**NCAs**	**DPs**
Training time:	Less than 1 month	☐	☒	☐
	Wall-clock days: 1.22 days; Hardware days: $3.92*10^{1}$ Nvidia GH200 days	☒	☐	☐
Amount of computation used for training:	10^{21} FLOPS	☐	☒	☐
	$1.1x10^{21}$ FLOPS	☒	☐	☐
Measurement methodology:	We estimated the number of FLOPS per token using the OpenAI scaling law paper. We then multiply that number by the number of tokens trained on.	☒	☒	☐

Energy consumption (during training and inference)		**AIO**	**NCAs**	**DPs**
Amount of energy used for training:	0.45 MWh	☒	☒	☐
Measurement methodology:	We train our model with Weights & Biases as logging system, this logs the power usage during training. We take the average power usage and then multiply by the compute time.	☒	☒	☐
Benchmarked amount of computation used for inference[1]:	Model runs comfortably on a 2 GB VRAM.	☒	☒	☐
Measurement methodology:	Measured using the vLLM inference runtime.	☒	☒	☐

[1] This item relates to energy consumption during inference, which makes up the “energy consumption of the model” (Annex XI, 2(e), AI Act) together with energy consumption during training. Since energy consumption during inference depends on more than just the model itself, the information required for this item is limited to relevant information depending only on the model, namely computational resources used for inference.