

Text Classification System using Natural Language Processing and Machine Learning with Generative Adversarial Networks*

Victor Sineglazov¹, Anna Lytovchenko²

¹ State University "Kyiv Aviation Institute", 1, Prospect Liubomyra Huzara, Kyiv, 03058, Ukraine

² National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Prospect Beresteiskyyi, Kyiv, 03056, Ukraine

Abstract

This work is devoted to develop a scalable multi-label classification system for Norwegian texts. We propose a novel architecture that fuses contextual embeddings from the NbAiLab/nb-bert-base model with a feature-level generative augmentation module based on f-VAEGAN-D2. By synthesizing label-conditioned embeddings for underrepresented classes and applying on-the-fly generative oversampling during classifier training, our method alleviates class imbalance and enhances recognition performance for both frequent and rare categories. We adapt the f-VAEGAN-D2 discriminator to operate on text embedding spaces, yielding substantial recall improvements on tail labels. It is offered practical guidelines for integration into municipal electronic document-routing systems that support both Bokmål and Nynorsk.

Keywords

Multi-label classification, Norwegian language, machine learning, large language models, BERT embeddings, f-VAEGAN-D2, class imbalance.

1. Introduction

Currently, with the growth of information volume, the problem of processing and classifying textual data remains undoubtedly relevant. It has gained wide popularity and is used for various tasks, such as spam classification, sentiment classification (sentiment analysis), and document categorization. In general, the classification process - namely assigning texts to a predefined set of classes - requires significant time and human resources when dealing with global tasks and large data volumes; therefore, using machine learning for text classification is a more appropriate option.

This is especially relevant for public authorities that receive thousands of emails per day which must be processed and routed [1]. For automatic distribution of emails by categories, it is necessary to analyze their content, identify key topics, and forward them to the email address of the relevant departments. Existing solutions based on manual processing or simple algorithms (such as keyword filters) are ineffective due to subjectivity, high time costs, and the growing variety of linguistic constructions in emails. Of special interest are the Scandinavian countries, in particular Norway, since according to the report by Statistisk sentralbyrå (SSB) [2] the share of Norwegian municipalities that do not use electronic tools, including email, fell from 37% (2018) to 6.5% (2022); 95.5% of municipalities in 2022 used email as the main channel of communication with citizens, which leads to additional load due to the large number of electronic requests. Norwegian language, having two official written standards and limited annotated corpora, poses a challenge for traditional machine learning methods applied to classification tasks.

Natural Language Processing (NLP) is a machine learning (ML) technology that enables computers to interpret, manipulate, and understand human language [3]. Programming within NLP combines linguistics and computer science with the aim of decoding the structure of language and

*ProfIT AI'25: 5th International Workshop of IT-professionals on Artificial Intelligence, October 15–17, 2025, Liverpool, UK

✉ svm@nau.edu.ua (V.Sineglazov); lytovchenko.anna@iit.kpi.ua (A. Lytovchenko)

ORCID 0000-0002-3297-9060 (V. Sineglazov); 0009-0002-6279-2843 (A. Lytovchenko)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

the rules of its use in order to detect, decompose into components, and extract meaningful information from text and speech [4]. By combining computational linguistics with statistical models, machine learning, and deep learning, NLP enables computers to recognize, analyze, and generate text and speech [5]. The field traces back to the Turing Test proposed by Alan Turing in the 1950s [7]. Subsequent milestones include the 1954 Georgetown experiment in machine translation [7], rule-based systems such as ELIZA in the 1960s [8], the rise of corpus-based statistical methods in the 1980s and 2000s (Penn Treebank, WordNet, SVMs, HMMs) [7], and the launch of Google Translate in 2006 [8]. From 2000 to 2010, ML and neural networks transformed NLP; today models like [26], GPT, and LLaMA achieve high accuracy across tasks, and the market is projected to reach USD 92.7 billion by 2028 [8]. In practice, approaches span rules-based NLP, statistical NLP, and deep-learning-based NLP.

Norwegian presents specific challenges for NLP. Two written standards – Bokmål ($\approx 85\text{--}90\%$) and Nynorsk ($\approx 10\text{--}15\%$) – differ in orthography, grammar, and lexicon, preventing a universal model without multi-corpus preparation [19]. The language is morphologically productive with extensive compounding (e.g., *høyhastighetstog*), which complicates tokenization [20]. Rich inflection, flexible word order, and numerous regional dialects further increase variability, impacting parsing and representation learning [19].

This research explores the integration of an intelligent multi-label text classification system for Norwegian using NLP, machine learning, and generative adversarial neural networks. The system is aimed at automatically determining to which category or categories an input text belongs. Special attention is paid to limited training data; we therefore apply a generative learning approach based on the f-VAEGAN-D2 framework, which augments the training corpus with high-quality synthetic examples.

2. Literature review

In the field of Natural Language Processing for text classification is commonly organized as a staged pipeline that converts raw messages into machine-interpretable features [9]. The literature describes a progression from tokenization, sentence and word segmentation, stop-word filtering, normalization, and vectorization to downstream classifiers ranging from linear models and ensembles to deep neural networks and transformers; ensemble and hybrid designs are well studied in this context [30]. For Norwegian – where two written standards (Bokmål and Nynorsk) coexist and morphological productivity is high – the quality of each stage has a measurable effect on final metrics, making preprocessing and representation learning particularly consequential [19], [20]. These choices can be framed as multi-criteria trade-offs among accuracy, robustness, and cost, for which formal multi-criteria optimization perspectives are relevant [33].

Research on tokenization for Norwegian addresses ambiguous periods (abbreviations, domains, decimals), hyphenated constructions, fixed expressions, and compound nouns. Practical systems combine rules, regular expressions, and machine learning, while modern pipelines favor subword approaches such as SentencePiece within AutoTokenizer, which remove fixed-vocabulary dependence and better handle compounding and orthographic variation between Bokmål and Nynorsk [21]. Preserving semantics at this earliest stage improves the fidelity of later vectorization and classification.

Stop-word filtering, normalization, and lemmatization are standard tools for reducing noise and vocabulary size. Off-the-shelf Norwegian stop-lists (e.g., in NLTK) are often a starting point but typically require domain adaptation. Normalization via stemming or lemmatization reduces type sparsity and stabilizes frequency statistics, which is useful for both inflection and compounding. These steps tend to improve efficiency and, when tuned to the domain, can improve effectiveness by sharpening the signal available to classifiers. The selection and topology of models used downstream are also influenced by foundational analyses of artificial neuron and network topologies [31].

Vectorization has shifted from Bag-of-Words and TF-IDF—simple, interpretable, but context-agnostic representations—to contextual embeddings that encode meaning as a function of surrounding tokens. BERT-style representations operate at the subtoken level, capture context, and preserve multi-word expressions, delivering higher accuracy on Norwegian classification tasks than sparse, high-dimensional count vectors. Contextualization is especially valuable where inflection and compounding would otherwise explode the vocabulary and obscure semantic relatedness across forms.

Classical classifiers remain relevant reference points. Naive Bayes and logistic regression are strong baselines for short texts but rely on assumptions—conditional independence and linear separability – that limit performance on longer sequences and multi-label settings [11]. Support Vector Machines (SVM) perform well with TF-IDF features and small training sets but are sensitive to kernel choice and regularization and do not scale gracefully to a large number of labels without reduction schemes [11]. Ensembles such as Random Forests and XGBoost capture nonlinearities and are robust to sparse or noisy inputs, yet, like linear models, they lack explicit modeling of word order and long-range dependencies.

Early neural approaches for text classification addressed these gaps by modeling sequential context. Recurrent Neural Networks (RNN) removed the independence assumption but suffered from vanishing and exploding gradients on long sequences. LSTM and GRU introduced gating mechanisms that significantly improved long-distance dependencies, at the expense of slower training and limited parallelism [12]. Convolutional Neural Networks (CNN) for text offered speed and the ability to learn local n-gram-like patterns useful for sentiment, toxicity, and stylistic cues, but they are less suited to capturing global discourse structure compared with attention-based models. When multiple, often conflicting, objectives arise—e.g., accuracy vs. latency vs. robustness—multi-criteria and evolutionary optimization methods, including genetic-algorithm-based conditional optimization, can guide model and threshold selection [32], [33]. Transformers fundamentally changed the state of the art. Architectures such as BERT, GPT, T5, and RoBERTa leverage self-attention to use full-sentence context [10]. BERT introduced bidirectional encoding and the [CLS] token as a document-level aggregate, which became a de facto standard for classification heads. For Norwegian, NB-BERT/NbAiLab variants adapted to Bokmål/Nynorsk consistently outperform classical methods on categorization tasks. In practice, however, scarcity of labeled data and severe label imbalance remain barriers, leading to overfitting on frequent classes and low recall in the long tail - effects that are amplified in multi-label regimes.

To reduce reliance on large labeled corpora, several works explore adversarial generation for text [13]. SeqGAN casts the generator as a reinforcement-learning agent that receives a reward from the discriminator after sequence completion, enabling GANs to operate over discrete tokens [14]. TextGAN introduces a feature-matching loss that encourages the generator to align distributions of discriminator-level features between real and generated sentences [15], [27]. MaliGAN reduces gradient variance by reparameterizing rewards, improving training stability [16]. RankGAN replaces binary discrimination with pairwise ranking, which correlates better with graded text quality [17]. Despite these advances, such models emphasize generation rather than many-to-one classification and seldom incorporate label information explicitly during training.

Hybrid approaches combine the strengths of autoencoding and adversarial training. f-VAEGAN-D2 generates discriminative feature vectors in an embedding space and supports any-shot scenarios (zero-/few-shot) by pairing a conditional discriminator with an unconditional that improves the marginal feature distribution [18]. The presence of an encoder permits many-to-one usage that aligns with classification. In text adaptations, contextual vectors (e.g., the BERT [CLS] embedding) serve as inputs, and the generator synthesizes label-conditioned features to expand rare classes without duplicating real examples. Such synthetic feature-level oversampling tends to preserve semantics better than simple data-level heuristics such as EDA or back-translation and integrates naturally with multi-label optimization and per-label threshold calibration. For Norwegian’s dual standards, compounding, and dialectal variability, robust preprocessing and subword tokenization (SentencePiece) are necessary components [19] – [21]. Building on these

findings, our approach fuses NbAiLab/nb-bert-base with f-VAEGAN-D2 to target rare-label enrichment in a multi-label setting, addressing gaps left by prior work in handling imbalance and preserving class semantics during training.

3. Problem Statement

The problem is to build multi label text classification system for Norwegian under scarce annotation and label imbalance. The input corpus contains labeled samples $L = \{(t_i, y_i)\}_{i=1}^{N_L}$, where each text t_i is accompanied by a binary label vector $y_i \in \{0,1\}^K$ and unlabeled samples $U = \{t_j\}_{j=1}^{N_U}$. Texts are encoded with contextual features using NbAiLab/nb-bert-base[6]. The document embedding is taken from the [CLS] token (classification token), which is added at the beginning of the input sequence[28]. After passing through several transformer layers, this token aggregates contextual information about the entire text, as in:

$$x_{\text{real}} = \text{BERT}_{[\text{CLS}]}([t_1, t_2, \dots, t_n]). \quad (1)$$

The goal is to estimate the conditional probability $p(x|y)$ of the label vector y given the embedding x and learn a mapping, as in:

$$f_{\theta}: \mathbb{R}^{768} \rightarrow [0,1]^K, \quad (2)$$

$$\hat{y} = f_{\theta}(x), \quad (3)$$

with independent per-label thresholding $\hat{y}_k \geq \tau_k$ environments, where \hat{y}_k - the model-predicted probability that the document belongs to label k , τ_k - the threshold set for label k .

The function f_{θ} is a parameterized model that takes as input a 768-dimensional document embedding (the output of the BERT \[CLS] token) and returns K values in the range $[0,1]$. Each of these values represents the predicted probability that the document belongs to the corresponding label.

4. Method overview

4.1. Neural Network Model

The key idea is to use the generative model f-VAEGAN-D2 as a classifier booster and as a source of synthetic features (vector representations) for rare classes. This enables robust learning on imbalanced datasets typical of real-world classification of municipal email inquiries.

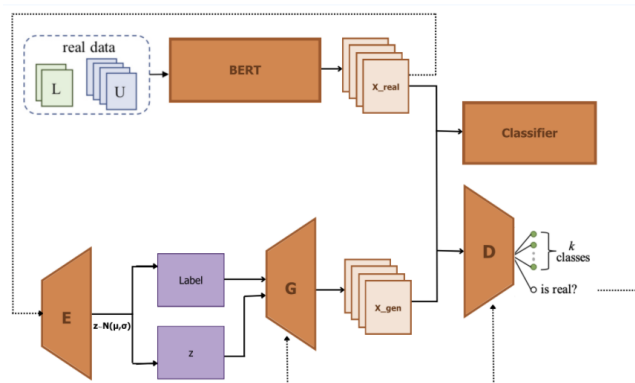


Figure 1: Topology of the proposed Neural Network Model

4.1.1. Encoder

The encoder approximates the posterior of a latent variable z given text embeddings and class labels. It maps $[x, y]$, where $\hat{x} \in \mathbb{R}^{768}$ is the BERT embedding and $y \in \{0, 1\}^K$ is the multi-label vector, to latent parameters $z \in \mathbb{R}^{64}$ with $q(z|x, y) \sim N(\mu, \sigma^2 I)$, where meaning the approximate posterior over z is modeled as a multivariate normal distribution whose mean $\mu(x, y)$ and diagonal covariance $\sigma^2(x, y)I$ are output by the encoder.

Architecture: one hidden layer with 128 units and two heads outputting μ and $\log \sigma^2$.
Reparameterization:

$$z = \mu(x, y) + \sigma(x, y) \odot \varepsilon, \varepsilon \sim N(0, I), \quad (4)$$

where $\mu(x, y)$ and $\sigma(x, y)$ are the mean and standard-deviation vectors output by the encoder for input (x, y) and $\varepsilon \sim N(0, I)$ is random noise drawn from the standard normal distribution.

Encoder loss includes (i) reconstruction MSE between original and reconstructed embeddings, and (ii) KL divergence between the approximate posterior $q(z|x, y)$ and the prior $p(z) \sim N(0, 1)$:

$$L_{enc} = MSE(x, \hat{x}) + \beta \cdot D_{KL}(q(z|x, y) || p(z)), \quad (5)$$

where \hat{x} is the reconstruction of the input embedding, β is a weighting coefficient that balances reconstruction (MSE) and latent-space regularization (encouraging proximity to $N(0, 1)$).

This yields a smooth, meaningful latent space suitable for reconstruction and feature generation for downstream classification.

4.1.2. Generator

The generator synthesizes text embeddings. It is a two-layer MLP that takes (z, y) with $z \in \mathbb{R}^{64}$ and $y_i \in \{0, 1\}^K$.

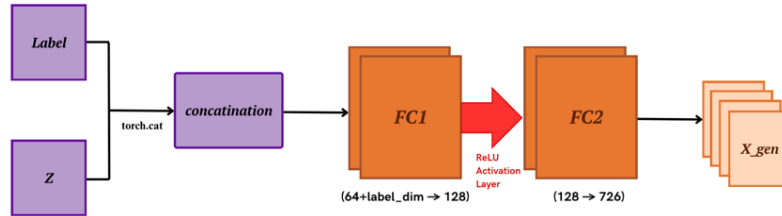


Figure 2: Architecture of the Generator of the proposed Neural Network Model

Layer 1: 128 units, ReLU. Layer 2: output $\hat{x} \in \mathbb{R}^{768}$ matching BERT's embedding size.

Models:

- reconstruction: z from the encoder to recover $x \approx \hat{x}$;
- generation: $z \sim N(0, 1)$ with arbitrary y to synthesize embeddings for zero-/few-shot support, especially for rare classes. The synthetic samples augment the training set prior to classification.

Generator loss:

$$L_G = L_{adv} + \lambda_{rec} \cdot L_{MSE} + \lambda_{KL} \cdot L_{KL} + \lambda_{FM} \cdot L_{FM}, \quad (6)$$

where L_{adv} is the adversarial loss, L_{MSE} reconstruction error, as in (7) between real and generated embeddings, L_{KL} latent regularization via the encoder, as in (8), and L_{FM} feature

matching between intermediate discriminator features $\phi(\cdot)$, as in(9), and $\lambda_{rec}, \lambda_{KL}, \lambda_{FM} \geq 0$ weight the respective terms.

$$L_{MSE} = \frac{\|x - \hat{x}\|_2^2}{d}, \quad (7)$$

where d is a is the embedding dimensionality and $d = 768.4$

$$L_{KL} = D_{KL}(z|x, y) \| N(0, I), \quad (8)$$

$$L_{FM} = \|\phi(\hat{x}, y) - \phi(x, y)\|_1. \quad (9)$$

4.1.3. Discriminator

The discriminator solves two tasks:

1. Adversarial discrimination between real x and synthetic $G(z, y)$
2. Feature matching to stabilize training by aligning hidden-layer statistics

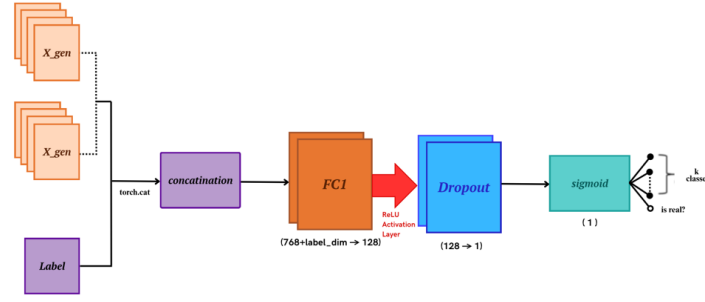


Figure 3: Architecture of the Discriminator of the proposed Neural Network Model

Input is the concatenation of an embedding and its label; processing follows[29]:

$$h_1 = \text{ReLU}(W_1 h_1 + b_1), \quad (10)$$

$$h_1 = \text{Dropout}(h_1, p=0.4), \quad (11)$$

$$D(x, y) = \sigma(W_2 h_1 + b_2), \quad (12)$$

where σ is the sigmoid, $D(x, y) \in (0, 1)$ is the probability that (x, y) is real, W_i - weight matrix, b_i is the bias, which yields the hidden representations before the nonlinearity. The hidden activation h_1 is also returned for feature matching.

Total discriminator loss combines:

- Adversarial loss (binary cross-entropy), as in:

$$L_{adv} = -E_{(x, y) \sim p}[\log D(x, y)] - E_{z \sim N(0, 1), y \sim p_y}[\log(1 - D(G(z, y), y))], \quad (13)$$

where $E_{(x, y) \sim p}[\cdot]$ is expectation over real pairs (x, y) , $E_{z \sim N(0, 1), y \sim p_y}[\cdot]$ is expectation over latent noise z and sampled labels y .

- Feature matching comparing hidden means:

$$L_{FM} = \|E_x[h(x, y)] - E_z[h(G(z, y), y)]\|^2, \quad (14)$$

where h is the discriminator's hidden feature vector.

The hidden vector h is also returned to compute L_{FM} in the generator.

4.1.4. Classifier

The classifier is a three-layer MLP (384→192→output) trained on the expanded dataset (original and synthetic embeddings). Each class uses its own decision threshold optimized for F1.

Table 1

Classifier Architecture

Layer	Input size	Output size	Activation	Parameters
FC1	[64, 768]	[64, 384]	ReLU	295296
Dropout(p=0.4)	[64,384]	[64, 384]	-	0
FC2	[64, 384]	[64, 192]	ReLU	73920
Dropout	[64, 192]	[64, 192]	-	0
FC3	[64, 384]	[64, 192]	ReLU	4825

4.2. GANs Training Problems

4.2.1. Vanishing Gradient Problem

When the discriminator D becomes too accurate early, generator G receives almost no learning signal. With the “saturating” generator objective:

$$L_G = E_{z \sim p_z(z)}[\log(1 - D(G(z, y), y))], \quad (15)$$

we get $L_G \rightarrow 0$ and $\nabla_{\theta} L_G \rightarrow 0$, so training stalls [24]. To stabilize gradients, use the Wasserstein objective that optimizes Earth Mover’s Distance [24]:

$$L_{WGAN} = E_{x \sim p_{data}}[D(x, y)] - E_{z \sim p_z}[D(G(z, y), y)] \quad (16)$$

4.2.2. Mode collapse

Mode collapse occurs when G outputs a few patterns that fool D but do not cover the data distribution [23]:

$$G(z, y) \approx x y, \quad \forall z \sim N(0, I) \quad (17)$$

The Wasserstein objective reduces collapse because minimizing a distance, not a log-probability, continues to provide usable updates even when D separates modes well. An equivalent form is

$$L = E_{x \sim p_{data}}[D(x, y)] - E_{z \sim p_z}[D(G(z, y), y)], \quad (18)$$

which preserves non-degenerate gradients when diversity drops.

4.2.3. Non-Convergence

In this case, a small Gaussian noise is added to the real data to make the task slightly harder for D and give G more time to adapt. Penalties on excessively large weights in D are also applied, which makes its task harder and helps preserve competition [22].

5. Results

The algorithm was tested on a manually collected dataset of the emails in Norwegian language, collected from the internal correspondence of the Nord-Aurdal kommune. Each email could belong to one or more of 17 predefined municipal categories (Aurdal omsorgssenter, Barnehave

virksomhetsleder, Brannvesenet, Eiendom, Fagernes legesenter, Helse og omsorg, Helsesøstertjenesten, Interkommunal barneverntjenest, Kultur, Miljø og Næring, Nord-Aurdal folkebibliotek, PP-tjeneste for Valdres, Regnskap, Skole virksomhetsleder, Teknisk, Økonomi), forming a multi-label classification task.

The dataset was split into 70/15/15, where 70% is training, 15% is validation and 15% is test subsets with three random initializations (7, 42, 2025) ensembled by probability averaging.

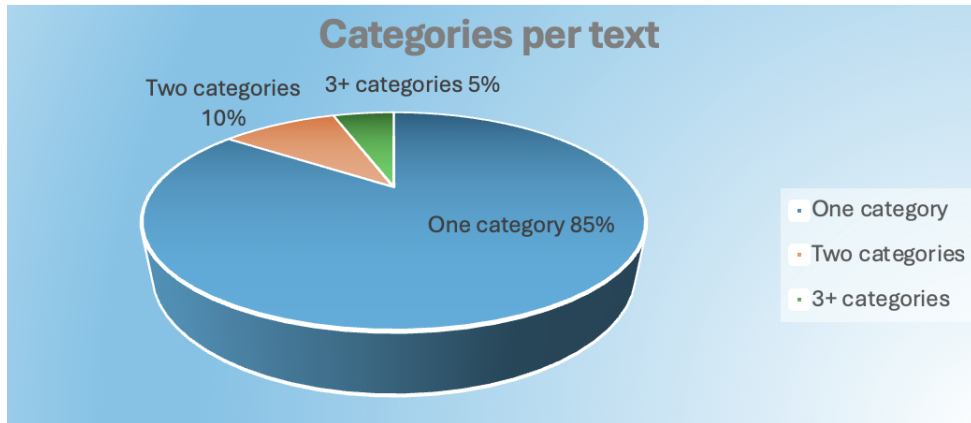


Figure 4: Number of categories per text for the collected dataset

One of the key issues identified during the dataset analysis is a significant imbalance between categories: several classes (such as Kultur and Regnskap) are represented by fewer than 20 examples. This situation prevents stable classifier training—the model tends either to completely ignore rare labels or to overfit on noisy patterns. For each underrepresented class, 1,000 new embeddings were synthesized by passing a random latent vector through the generator along with the corresponding one-hot label representation. The generated embeddings were integrated into the training set, augmenting the real examples. The classification model (an ensemble of MLPs) was then trained on this extended dataset and was trained for 50 epochs with a batch size of 32 using the optimizer (learning rate = $1e-3$). GAN was trained for 12 epochs with batch size 64 using Adam (lr = $1e-3$) for Encoder, Generator, Discriminator. The loss combined BCE (adversarial), MSE (reconstruction), KL-divergence, and feature matching and gradients were batch-averaged. Three random initializations (7, 42, 2025) were ensembled by probability averaging.

To evaluate the performance of the proposed neural network model, several standard metrics are used:

- Precision
- Recall
- F1 score. Two forms of F1-score were analyzed separately:
 1. Micro-F1 is calculated globally across all labels at once (i.e., all TP, FP, and FN are summed). It is sensitive to classes with a large number of examples.
 2. Macro-F1 is the average F1-score computed separately for each class. Unlike Micro-F1, it is not affected by class frequency and better reflects performance on rare categories.

To evaluate the reliability of threshold-based predictions, calibration metrics are used:

- AUPRC (Area Under Precision–Recall Curve)
- Brier score (mean squared error between predicted probabilities and actual labels)

This allows us to clearly compare model versions and ensure the system works reliably in practical scenarios.

Table 2

Metrics for evaluation

Metric	Value
Precision	0.92
Recall	0.744
F1	Micro-F1 0.823
	Macro-F1 0.68
AUPRC	0.832
Brier	0.17

Per-label thresholds is tuned on the validation split to maximize Macro-F1 and then froze them for testing. An ablation study (no augmentation vs class-conditional augmentation) showed consistent gains on tail labels when synthetic, label-conditioned embeddings were included. Precision–Recall curves (micro and macro) further confirm robustness under class imbalance. Calibration was evaluated using mean AUPRC = 0.83 and mean Brier score = 0.17 with good alignment between predicted probabilities and true label frequencies.

6. Conclusions

In this work was presented system that combines BERT-based embeddings with generative data augmentation for multi-label classification for texts in Norwegian Language. The pipeline ensures normalized and denoised text, cleaned from noise and redundant morphology.

Text vectorization is carried out using the NbAiLab/nb-bert-base model, which produces deep, contextualized embeddings. To address class imbalance, used an f-VAEGAN-D2 architecture to synthesize additional embeddings for rare categories, preserving the latent-space structure and enhancing classification quality.

Inference is performed using an ensemble of neural networks trained on both real and synthetic embeddings, with per-label probability thresholds optimized for each category. Architectural choices, regularization techniques, and a carefully designed training regimen prevent common GAN-related failures—gradient vanishing, unstable convergence, and mode collapse—even in the challenging setting of multi-label text classification.

Evaluation on test dataset by macro-F1 and micro-F1(0.823 and 0.68, respectively) confirms that overall performance improved and rare-class accuracy rose, reducing neglect of underrepresented labels. A mean AUPRC of 0.83 and Brier score of 0.17 indicate strong calibration. Per-label thresholds and ensemble inference ensured stable, accurate detection of rare categories. The architecture therefore shows strong potential for classification.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 in order to: Grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] “How many work emails is too many?”, *The Guardian*, Apr. 8, 2019. Available: <https://www.theguardian.com/technology/shortcuts/2019/apr/08/how-many-work-emails-is-too-many>.
- [2] “Digitalisation in the Norwegian municipalities: Development from 2018 to 2022,” *Statistics Norway (SSB)*. Available: <https://www.ssb.no/en/teknologi-og-innovasjon/informasjons-og-kommunikasjonsteknologi-ikt/artikler/digitalisation-in-the-norwegian-municipalities-development-from-2018-to-2022>.
- [3] “Natural language processing”, *Wikipedia*. Available: https://en.wikipedia.org/wiki/Natural_language_processing.
- [4] “Natural Language Processing”, *Engati Glossary*. Available: <https://www.engati.com/glossary/natural-language-processing>.
- [5] “What is NLP?”, *IBM Think*. Available: <https://www.ibm.com/think/topics/natural-language-processing>.
- [6] NbAiLab, “NbAiLab/nb-bert-base,” *Hugging Face*, Accessed: Aug. 31, 2025. [Online]. Available: <https://huggingface.co/NbAiLab/nb-bert-base>.
- [7] R. S. T. Lee, *Natural Language Processing: A Textbook with Python Implementation*. Cham: Springer, 2023, 437 p.
- [8] “Evolution of NLP: From Past Limitations to Modern Capabilities,” *Medium*. Available: https://medium.com/@social_65128/evolution-of-nlp-from-past-limitations-to-modern-capabilities-6dc1505faeb6.
- [9] “Understanding the NLP Pipeline: A Comprehensive Guide,” *Medium*. Available: https://medium.com/@asjad_ali/understanding-the-nlp-pipeline-a-comprehensive-guide-828b2b3cd4e2.
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv*, 2019. doi: 10.48550/arXiv.1810.04805.
- [11] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Computing Surveys*, 2022.
- [12] Y. Goldberg, “A Primer on Neural Network Models for Natural Language Processing,” *Journal of Artificial Intelligence Research*, 2016.
- [13] J. Langr and V. Bok, *GANs in Action: Deep Learning with Generative Adversarial Networks*. Shelter Island, NY: Manning Publications, 2019, 240 p.
- [14] L. Yu, W. Zhang, J. Wang, and Y. Yu, “SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient,” *arXiv*, 2017. Available: <https://arxiv.org/abs/1609.05473>.
- [15] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, and L. Carin, “Adversarial Feature Matching for Text Generation,” *arXiv*, 2017. Available: <https://arxiv.org/abs/1706.03850>.
- [16] T. Che, Y. Li, R. Zhang, R. D. Hjelm, W. Li, Y. Song, and Y. Bengio, “Maximum-Likelihood Augmented Discrete Generative Adversarial Networks,” *arXiv*, 2017. Available: <https://arxiv.org/abs/1702.07983>.
- [17] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun, “Adversarial Ranking for Language Generation,” *arXiv*, 2017. Available: <https://arxiv.org/abs/1705.11001>.
- [18] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, “f-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning,” in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 10275–10284. doi: 10.1109/CVPR.2019.01053.
- [19] Språkrådet, “Om norsk språk og standarder,” 2023. Available: <https://www.sprakradet.no>.
- [20] L. Øvrelid and E. Velldal, “Syntactic variation and parsing of Norwegian,” in *Proceedings of VarDial*, 2020.
- [21] Teodorescu M.H. Machine Learning Methods for Strategy Research. HBS Working Paper 18-011. Harvard Business School, 2017. 59 p.

- [22] “GAN — Why it is so hard to train Generative Adversarial Networks!,” *Medium*. Available: <https://jonathan-hui.medium.com/gan-why-it-is-so-hard-to-train-generative-advisory-networks-819a86b3750b>.
- [23] M. Zamorski, A. Zdobylak, M. Zięba, and J. Świątek, “Generative Adversarial Networks: recent developments,” *arXiv*, 2019. doi: 10.48550/arXiv.1903.12266.
- [24] M. M. Saad, R. O’Reilly, and M. H. Rehmani, “A Survey on Training Challenges in Generative Adversarial Networks for Biomedical Image Analysis,” *arXiv*, 2022. doi: 10.48550/arXiv.2201.07646.
- [25] “Evaluation Metrics in Machine Learning,” *GeeksforGeeks*. Available: <https://www.geeksforgeeks.org/machine-learning/metrics-for-machine-learning-model/>.
- [26] 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, Minneapolis, Minnesota, 2019, pp. 4171–4186.
- [27] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, L. Carin, Adversarial feature matching for text generation, in: 34th International Conference on Machine Learning, 2017, pp. 4006–4015.
- [28] W. Nie, N. Narodytska, A. Patel, Relgan: Relational generative adversarial networks for text generation, in: 7th International Conference on Learning Representations, 2019.
- [29] L. Chen, S. Dai, C. Tao, H. Zhang, Z. Gan, D. Shen, Y. Zhang, G. Wang, R. Zhang, L. Carin, Adversarial text generation via feature-mover’s distance, in: Advances in Neural Information Processing Systems, 2018, pp. 4666–4677.
- [30] Sineglazov, V., Kot, A. Design of Hybrid Neural Networks of the Ensemble Structure Eastern European Journal of Enterprise Technologies, 2021, 1, pp. 31–45.
- [31] Zgurovsky, M., Sineglazov, V., Chumachenko, E. Classification and Analysis Topologies Known Artificial Neurons and Neural Networks Studies in Computational Intelligence, 2021, 904, pp. 1–58.
- [32] Zgurovsky, M., Sineglazov, V., Chumachenko, E. Classification and Analysis of Multicriteria Optimization Methods Studies in Computational Intelligence, 2021, 904, pp. 59–174.
- [33] Sineglazov, V.M., Riazanovskiy, K.D., Chumachenko, O.I. Multicriteria conditional optimization based on genetic algorithms System Research and Information Technologies, 2020, 2020(3), pp. 89–104.