Open Access Library Journal
2026, Volume 13, e15181
ISSN Online: 2333-9721
ISSN Print: 2333-9705

Topic Classification of Fulani Texts Using Neural Networks

Mamadou Diaye

Department of Computer Science, Hubei University of Technology, Wuhan, China
Email: diayemamadou21@gmail.com

How to cite this paper: Diaye, M. (2026) Topic Classification of Fulani Texts Using Neural Networks. *Open Access Library Journal*, **13**: e15181.
https://doi.org/10.4236/oalib.1115181

Received: March 16, 2026
Accepted: November 22, 2026
Published: November 25, 2026

Copyright © 2026 by author(s) and Open Access Library Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

Despite recent advances in Natural Language Processing (NLP), many African languages remain severely under-resourced, limiting their inclusion in modern language technologies. Fulani (Fulfulde/Pulaar/Fula) is one such language, for which there is a notable lack of benchmark datasets and empirical studies. This paper addresses the problem of automatic topic classification of Fulani texts using neural and Transformer-based models. To this end, we construct and manually annotate a Fulani text corpus spanning five topical categories: politics, health, education, religion, and culture. We investigate a deep learning framework that integrates distributed word representations and evaluates both classical neural architectures—namely Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Bidirectional LSTM (BiLSTM)—as well as Transformer-based models, including multilingual pre-trained Transformers. Experimental results, assessed using accuracy, precision, recall, and F1-score, show that while recurrent and convolutional models achieve competitive performance, Transformer-based models consistently provide superior classification accuracy in the low-resource setting. This work establishes one of the first systematic benchmarks for Fulani topic classification and provides a foundation for future NLP research on low-resource African languages.

Subject Areas

Network Modeling and Simulation

Keywords

Fulani Language, Topic Classification, Low-Resource NLP, Text Classification, Transformer Models

1. Introduction

1.1. Linguistic and Historical Background of the Fula Language

Fula, also known by alternative names such as Fulah, Ful, Fulfulde, Fulani, Fulbe, Peul, Pular, and Pulaar, belongs to the Niger-Congo language family, within the Atlantic branch, specifically the Northern Atlantic subgroup, alongside languages such as Wolof and Serer. It is best described as a macrolanguage or dialect continuum, rather than a single uniform language, due to the wide range of mutually related but sometimes not fully mutually intelligible varieties spoken across a vast geographical area of Africa. Table 1 summarizes the alternative names and genetic classification of the Fula language.

Table 1. Alternative names and genetic classification of Fula.

Category	Description
Alternative names	Fulah, Ful, Fulfulde, Fulani, Fulbe, Peul, Pular, Pulaar
Language family	Niger-Congo
Branch	Atlantic
Subgroup	Northern Atlantic
Related languages	Wolof, Serer

Historically, Fula is distinguished by its exceptionally wide geographic distribution, extending from the Atlantic coast of Senegal to the savannas of East Africa. This extensive spread is closely linked to the traditionally nomadic and pastoral lifestyle of the Fulbe people. Unlike most other Atlantic languages, which are largely confined to the West African coastal region, Fula stretches far inland and is spoken across at least sixteen countries. The Fula-speaking area is not geographically continuous, but rather fragmented by regions in which numerous other languages are spoken. It is estimated that approximately 18 million people speak a Fula dialect across multiple countries in West and Central Africa. Major speaker populations are found in Guinea, Senegal, Nigeria, Mali, Cameroon, Niger, and Burkina Faso, among others. Table 2 presents an overview of estimated speaker populations by country, highlighting the wide geographical and demographic spread of the language.

The language exhibits significant dialectal variation, forming a classic dialect continuum in which neighboring varieties are generally mutually intelligible, while more distant varieties may not be. Several major varieties are commonly identified in the literature, each associated with specific regions and speaker communities. Table 3 lists the main Fula varieties and their approximate speaker populations. From a historical perspective, the earliest records of Fula date back to the 12th - 14th centuries, when Fula words were documented by Arabic travellers in the Western Sudan, while the first European records appeared in the 17th century. The language has traditionally been written using the Arabic script (Ajami), a

practice that continues in some regions, although the Latin script is now the dominant writing system in most countries. The modern orthography includes conventions such as the doubling of vowels and consonants to mark length, the use of digraphs for prenasalised consonants (e.g., mb, nd, ng), and the apostrophe to represent the glottal stop, with some regional variations in letter usage.

Table 2. Estimated number of Fula speakers by country.

Country	Estimated speakers
Guinea	4,200,000
Senegal	3,700,000
Nigeria	2,400,000
Mali	1,600,000
Cameroon	1,500,000
Niger	1,300,000
Burkina Faso	1,100,000
Benin	350,000
Gambia	350,000
Guinea-Bissau	300,000
Sierra Leone	270,000
Central African Republic	230,000
Chad	200,000
Mauritania	200,000
Sudan	130,000
Togo	70,000

Table 3. Major Fula varieties and approximate speaker populations.

Variety	Main regions	Approx. speakers
Pulaar (Peul)	Senegal, Gambia, Guinea-Bissau, Mali, Mauritania	4.5 - 5 M
Pular (Futa Jalon)	Guinea, Senegal, Sierra Leone	4.5 - 5 M
Maasina Fulfulde	Mali	1.1 M
Borgu Fulfulde	Benin, Togo	400 K
Western Niger Fulfulde	Niger, Burkina Faso	1.5 M
Central-Eastern Niger Fulfulde	Niger	650 K
Nigerian Fulfulde	Nigeria	2 M
Adamawa Fulfulde	Cameroon, Chad, Sudan	2 M
Bagirmi Fulfulde	Central African Republic, Chad	250 K

Linguistically, Fula displays several features that are particularly relevant for computational processing. Phonologically, it has a system of five basic vowels that can be short or long and a rich consonant inventory characterised by voiceless, voiced, laryngealised, and prenasalised series. Notably, unlike most Niger-Congo languages, Fula and many other Atlantic languages lack lexical tone. Morphologically, Fula is characterised by an elaborate noun class system, typically comprising between 20 and 25 classes depending on the dialect, with number marking involving both suffix changes and initial consonant mutation. This mutation system, in which consonants alternate between continuant, stop, and nasal grades, introduces substantial surface variation in word forms.

The verbal system is organized mainly around voice and aspect rather than tense, with distinctions such as perfective and imperfective, and includes active, middle, and passive voices, as well as various derivational extensions. Syntactically, the neutral word order is Subject-Verb-Object, and the language is head-initial, with modifiers following the head noun. Fula also makes extensive use of prepositions and employs specific focus constructions. Lexically, the language contains loanwords from Arabic, French, and English, reflecting historical contact and sociolinguistic influences. These linguistic characteristics—particularly the rich morphology, consonant mutation, orthographic variation, and dialectal diversity—pose significant challenges for Natural Language Processing. At the same time, they make Fula an important and representative case study for investigating text classification in low-resource and morphologically complex languages, thereby justifying its selection as the focus of this study.

1.2. Challenges of NLP for Low-Resource Languages

Natural language processing research remains concentrated on a small number of high-resource languages, while most African languages are severely under-represented [1] [2]. Developing NLP systems for low-resource languages presents several fundamental challenges. The most prominent of these is data scarcity. Unlike high-resource languages, which benefit from massive annotated corpora and large-scale web data, low-resource languages typically lack sufficient labeled datasets for supervised learning. This limitation directly affects the performance and generalization ability of data-driven models, particularly deep neural networks, which are known to require substantial amounts of training data.

Another major challenge is the lack of linguistic tools and benchmarks. For many low-resource languages, essential resources such as tokenizers, part-of-speech taggers, morphological analyzers, and standardized evaluation datasets are either unavailable or only partially developed. The absence of such tools makes it difficult to build reproducible NLP pipelines and to fairly compare different approaches. As a result, progress in these languages is often fragmented and difficult to measure against established standards. In the case of Fulani, additional linguistic complexities arise from morphological richness and orthographic variation. Fulani exhibits a relatively complex morphological structure, including affixation

and agreement patterns, which can increase data sparsity and complicate text representation. Furthermore, the language is written in multiple orthographic forms and is influenced by regional and dialectal variations, leading to inconsistencies in spelling and word forms across different sources. These characteristics pose further challenges for text preprocessing, vocabulary construction, and model generalization, and they highlight the need for robust modeling approaches that can cope with limited and noisy data.

1.3. Problem Statement and Research Objectives

Given the limited availability of computational resources and empirical studies for Fulani, there is currently no established benchmark for automatic topic classification in this language. This gap restricts both practical applications and further academic research on Fulani NLP. The central problem addressed in this study is therefore the development and evaluation of an automatic topic classification system for Fulani texts using modern deep learning techniques. Specifically, this work aims to investigate how effectively neural network-based and Transformer-based models can classify Fulani texts into predefined topical categories under low-resource conditions. To achieve this goal, the study pursues the following research objectives: 1) to construct and annotate a Fulani text corpus suitable for topic classification; 2) to design a preprocessing and representation pipeline tailored to the linguistic characteristics of Fulani; 3) to implement and compare several neural architectures, including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and selected Transformer-based models; 4) to evaluate these models using standard classification metrics in order to establish reliable baseline results.

This study makes several contributions to the field of NLP for low-resource languages, with a specific focus on Fulani. First, it presents the construction and annotation of a labeled Fulani text dataset for topic classification, addressing the lack of publicly available resources for this language. Second, it provides a systematic comparative evaluation of classical neural network models and modern Transformer-based models for Fulani text classification, offering insights into their relative strengths in a low-resource setting. Third, it establishes baseline experimental results that can serve as reference points for future research on Fulani and related African languages. Together, these contributions aim to facilitate further development of language technologies for Fulani and to encourage broader inclusion of low-resource languages in NLP research.

The remainder of this paper is organized as follows. Section 2 reviews related work on text classification, neural network models, Transformer-based approaches, and NLP for low-resource African languages. Section 3 describes the dataset construction, annotation, and preprocessing steps applied to the Fulani texts. Section 4 presents the methodology, including the model architectures and training objective. Section 5 details the experimental setup and evaluation protocol. Section 6 reports the empirical results and an error analysis. Section 7 discusses the find-

ings and the limitations of the study. Finally, Section 8 concludes the paper and suggests directions for future work.

2. Literature Review

2.1. Deep Learning for Text Classification

Text classification is one of the most fundamental tasks in NLP and has been widely studied across many languages and domains. Early approaches relied on classical machine learning with hand-crafted features, whereas deep learning architectures have become dominant because they learn hierarchical features directly from text without extensive manual feature engineering. Convolutional Neural Networks were shown to be effective for sentence- and character-level classification [3] [4], while recurrent architectures such as the Long Short-Term Memory (LSTM) network [5] and its bidirectional extension (BiLSTM) [6] were designed to model sequential dependencies. In practice, CNNs capture local n-gram features efficiently [3], whereas LSTM and BiLSTM encoders are better suited to modelling longer-range dependencies in sequential data [5] [6].

Although these models perform well in languages with large amounts of data (e.g., English, Chinese), their effectiveness degrades when applied to languages with limited linguistic resources, because of data scarcity and the lack of annotated corpora. Surveys of low-resource NLP emphasize that the absence of sufficient labelled text limits the training of complex neural architectures and motivates techniques such as data augmentation, transfer learning, and pretraining [2].

2.2. Transformer Models and Pretrained Language Representations

The Transformer architecture [7] and pretrained language models such as BERT (Bidirectional Encoder Representations from Transformers) [8] have significantly advanced the state of the art in many NLP tasks, including classification. Pretrained Transformers are trained on large corpora with self-supervised objectives and subsequently fine-tuned for specific downstream tasks, which typically yields large improvements over models trained from scratch. For low-resourced languages, Ogueji *et al.* [9] showed that multilingual pretrained models trained on a comparatively small amount of text covering several African languages can be competitive on downstream tasks. Large-scale multilingual encoders such as XLM-R [10], pretrained on text in 100 languages, and earlier cross-lingual pretraining work [11] have further demonstrated strong cross-lingual transfer, which makes them attractive for text classification in languages with sparse resources.

2.3. NLP for Low-Resource and African Languages

Despite advances in NLP, the majority of research has focused on high-resource languages, with African languages remaining largely under-represented [1] [12]. Community-driven initiatives have shown that corpus creation is a foundational step for progress: MasakhaNER, for example, built named entity recognition re-

sources for ten African languages through participatory research [13].

Recent work has also applied Transformers to African-language text classification. The AfriSenti shared task at SemEval-2023 covered sentiment analysis in 14 African languages, and many participating systems relied on fine-tuned multilingual or Africa-centric pretrained encoders [14]. Multilingual adaptive fine-tuning of XLM-R on African languages improved downstream tasks, including news topic classification and sentiment analysis, over the unadapted model [15]. Africa-centric pretrained models such as AfroLM [16] and SERENGETI [17] extend this approach to dozens and hundreds of African languages, respectively. Most closely related to our task, MasakhaNEWS provides a news topic classification benchmark for 16 African languages and compares fine-tuning and few-shot approaches [18].

Despite the growing interest in NLP for African languages, work on the Fulani language remains extremely limited. Fulfulde has appeared mainly as one of many languages in large multilingual resources such as machine translation systems [19], and, to the best of our knowledge, dedicated benchmarks and systematic studies of Fulani text classification remain absent from the literature. This study aims to fill this gap by constructing a labeled Fulani text corpus and evaluating both neural network and Transformer-based models for topic classification. The results are intended to establish baseline benchmarks that can support future NLP research on Fulani and other similarly under-resourced languages.

3. Dataset and Preprocessing

3.1. Data Collection and Corpus Composition

The Fulani (Fulfulde/Pulaar/Fula) corpus used in this study was constructed from scratch due to the absence of publicly available, standardized datasets for topic classification in this language. Texts were collected from multiple publicly accessible and community-driven digital sources, including online news portals, educational and government websites, religious and cultural blogs, Facebook groups dedicated to Fula language and culture, and transcripts of Fula-language radio programmes. These sources were chosen to ensure topical diversity, genre variation, and authentic real-world language usage.

Data collection was performed through a combination of manual curation and semi-automated web scraping. For each source, relevant pages were identified using keyword-based queries in Fulani, followed by manual verification to ensure linguistic authenticity. Only documents written predominantly in Fulani were retained. Texts exhibiting heavy code-mixing with French, English, or Arabic, or containing excessive noise (e.g., advertisements, navigation menus, or boilerplate content), were excluded. This filtering step was essential to preserve the linguistic integrity of the corpus and to reduce spurious correlations during model training.

Collection proceeded in three parallel streams: web crawling of Fula-language portals, blogs, and community sites; manual curation of openly licensed educational and religious material; and public social-media posts. The sources are

drawn from Senegal, Guinea, Nigeria, and Mali. Quality filtering removed very short and very long documents, near-duplicates, documents with a high proportion of non-Fula tokens (more than 40% French or English words), and texts with low Fula language-identification probability. It reduced the roughly 5900 collected documents to the final corpus of 5300 (about 90% retained). Table 4 reports the composition of the raw collection by source type. Documents were collected between January to July. The corpus contains Latin-script text only. Because Fula is a dialect continuum (Section 1.1) and the corpus is treated as a single dataset, we note that its dialectal distribution was not formally audited: western varieties (Pulaar and Pular) are over-represented relative to eastern varieties such as Nigerian and Adamawa Fulfulde (Section 7.2).

Table 4. Corpus composition by source type (raw collection, before quality filtering).

Source type	Primary topics	Raw documents
Online news portals	Politics, Health, Education	1800
Educational and government websites	Health, Education	900
Religious and cultural blogs	Religion, Culture	1400
Social media (Facebook groups)	All categories	1200
Broadcast media transcripts	Politics, Culture	600
Total before filtering		5900
Final corpus after filtering		**5300**

Note: raw counts are approximate.

Each document in the dataset is stored in a structured format consisting of: 1) A unique document identifier; 2) The raw text content; 3) The normalized and tokenized version of the text; 4) The assigned topic label. This structure facilitates reproducibility, traceability, and flexible experimentation with different preprocessing pipelines and modeling approaches. Furthermore, duplicate and near-duplicate texts were detected and removed using similarity-based filtering before the data were split, to prevent data leakage between the training, validation, and test sets.

3.2. Data Annotation and Class Definition

The dataset was manually annotated into five high-level topic categories: politics, health, education, religion, and culture. These categories were selected based on their frequency in Fulani digital content, their societal relevance, and their clear semantic separability, which is important for reliable supervised learning. Annotation was carried out by native or proficient Fulani speakers following a detailed annotation guideline that defined each class, provided inclusion and exclusion criteria, and illustrated typical examples.

Each document was annotated by one primary annotator, and a randomly se-

lected 20% subset was double-annotated to measure reliability. Table 5 summarizes the setup. Cohen's kappa of 0.84 is conventionally interpreted as almost perfect agreement. Disagreements were concentrated in the Culture category (61% of all disagreements, although Culture is only 17.9% of the corpus) and were adjudicated by a third, senior annotator. Where a document spanned several topics, annotators assigned the category reflecting its primary communicative purpose, so each document carries a single label.

Table 5. Annotation setup and reliability.

Aspect	Details
Number of annotators	4 native or near-native Fula speakers, representing western (Pulaar/Pular) and eastern (Nigerian/Adamawa Fulfulde) varieties
Annotation guideline	Written guideline with class definitions, inclusion/exclusion criteria, examples, and a primary-purpose rule for multi-topic documents
Calibration	Three rounds of joint annotation on a 50-document pilot sample, completed once agreement exceeded kappa = 0.80
Double-annotated portion	1060 documents (20% of the corpus)
Agreement score	Cohen's kappa = 0.84 (raw agreement 91.3%)
Adjudication procedure	Disagreements resolved by a third, senior annotator whose decision was final (93 documents, fewer than 10% of the double-annotated set)

Table 6 reports the main statistics of the constructed dataset after preprocessing, including the number of samples per class, the average document length in tokens, and the number of unique words per class. These statistics highlight both the moderate scale of the dataset and its lexical diversity, which reflects the morphological richness and orthographic variation of Fulani.

Table 6. Dataset statistics (after preprocessing).

Class	# Samples	Avg. length (tokens)	Unique words
Politics	1200	120	5300
Health	1000	110	4800
Education	1100	115	5100
Religion	1050	130	5600
Culture	950	105	4500
Total	**5300**	**116**	**12,400**

3.3. Text Pre-Processing

Prior to model training, all texts were processed using a pipeline tailored to the linguistic characteristics of Fulani. The rules applied in the final experiments are listed below; the same rules were applied uniformly to the training, validation, and test sets to avoid distributional inconsistencies.

1) *Cleaning.* HTML tags (BeautifulSoup), URLs, emoji, and special characters were removed, and whitespace was normalized.

2) *Orthographic and encoding normalization.* A custom normalization table (rather than a generic Unicode normal form) standardizes the prenasalized consonant digraphs (mb, nd, nj, ng) and unifies glottal-stop representations to the ASCII apostrophe. This avoids corrupting the prenasalized digraphs and implosive-consonant marks that carry phonemic contrast. Text was lowercased for the CNN, LSTM, and BiLSTM models.

3) *Tokenization.* For the CNN, LSTM, and BiLSTM models, text was split at white space with punctuation separated, and word types occurring fewer than three times in the training set were mapped to a single UNK token. The Transformer-based models used their native subword tokenizers (WordPiece for mBERT and SentencePiece [20] for XLM-R).

4) *Stopword removal.* The 143-entry Fulani stopword list described below was removed for the neural models only. Transformer-based models were also tested with filtered input; the difference was marginal (below 0.3% accuracy), and the results in Section 6 use unfiltered input.

5) *Truncation and padding.* All sequences were truncated or padded to a maximum length of 256 tokens, using zero-embedding padding for the neural models and a [PAD] token with attention masks for the Transformer models. Most documents are shorter than this limit (mean length 116 tokens, **Table 6**).

6) *Embeddings.* The CNN, LSTM, and BiLSTM models share an identical embedding layer initialized with pretrained 300-dimensional FastText subword vectors for Fula [21] and trained together with the model. The Transformer-based models use their native embedding layers.

Table 7. Sample of the constructed Fulani stopword list.

Stopword	Stopword	Stopword	Stopword
ko	e	nde	o
e	mi	maa	um
to	i	hay	ndee

A key challenge in preprocessing Fulani texts is the lack of standardized NLP resources, including stopword lists. To address this gap, we constructed a new, task-specific Fulani stopword list as part of this study. The stopwords were identified through a combination of frequency analysis, linguistic inspection, and manual verification by native speakers. Words that occur very frequently across

topics but carry limited semantic content for topic discrimination—such as function words, pronouns, and common particles—were included in the list. This custom stopword list represents an original linguistic resource contributed by this work and helps reduce noise and improve model focus on semantically informative tokens. The final list contains 143 entries; a small sample is shown in Table 7.

4. Methodology

4.1. Overall Model Architecture

The proposed model implements a unified and modular end-to-end topic classification framework for Fulani texts. The architecture, shown in Figure 1, follows a standard but carefully engineered NLP pipeline consisting of five main stages: 1) Input acquisition; 2) Preprocessing; 3) Representation learning (embedding); 4) Encoder model; 5) Classification layer.

Given a raw Fulani document, the text is first passed through the preprocessing module described in Section 3.3, which performs normalization, tokenization, optional stopword filtering, and sequence length standardization via padding and truncation. The resulting token sequence is then mapped to a dense vector representation using either a trainable or pretrained embedding layer (for neural models) or the native subword embedding mechanism of a Transformer-based model.

These vector sequences are subsequently processed by one of the selected encoders: a Convolutional Neural Network (CNN), a Long Short-Term Memory network (LSTM), a Bidirectional LSTM (BiLSTM), or a Transformer-based encoder (mBERT or XLM-R). The encoder produces a fixed-dimensional document representation, which is fed into a fully connected classification head followed by a softmax layer to produce a probability distribution over the predefined topic classes. The predicted class corresponds to the maximum posterior probability. This modular design enables a controlled and fair comparison between classical neural architectures and modern Transformer-based models under the same dataset, training protocol, and evaluation settings.

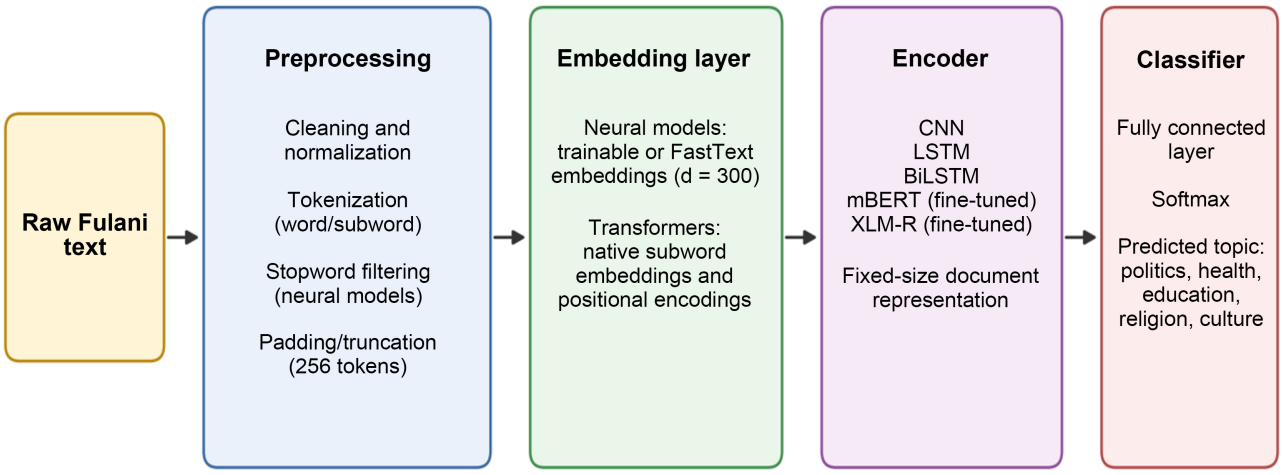


Figure 1. Overall architecture of the proposed Fulani topic classification system.

4.2. Text Representation and Embedding Layer

Two distinct representation strategies are adopted depending on the model family. For the neural network models (CNN, LSTM, and BiLSTM), each token is mapped to a dense vector pretrained 300-dimensional FastText subword embeddings [21]. Subword-based representations are particularly suitable for low-resource and morphologically rich languages such as Fulani, as they mitigate data sparsity by sharing representations across related word forms arising from inflection, derivation, and orthographic variation. Let a document be represented as a sequence of tokens ($w_1, w_2, \cdots, w_T$). Each token w_i is mapped to a vector $e_i \in \mathbb{R}^d$, forming an embedding matrix $E \in \mathbb{R}^{T \times d}$, which serves as input to the encoder.

For Transformer-based models such as mBERT and XLM-R, the input is first segmented using the model's native subword tokenization scheme (WordPiece or SentencePiece [20]). The embedding layer and positional encodings are integral components of the pretrained model, and contextualized token representations are produced internally during fine-tuning.

4.3. Convolutional Neural Network (CNN)

The CNN-based classifier operates on the embedding matrix $E \in \mathbb{R}^{T \times d}$ and applies multiple one-dimensional convolutional filters with varying kernel widths $k \in \{3,4,5\}$ to capture local n-gram features, following [3]. Each convolution operation is followed by a rectified linear unit (ReLU) activation function and a max-over-time pooling operation to extract the most salient feature from each feature map. The pooled features from all filters are concatenated to form a fixed-dimensional vector, which is then passed through a fully connected layer and a softmax classifier. This architecture is computationally efficient and effective at modeling local discriminative patterns that are indicative of topical content.

4.4. Long Short-Term Memory (LSTM)

The LSTM model [5] processes the input sequence sequentially and maintains a hidden state that captures long-range dependencies through gated memory cells. Given the sequence of embeddings ($e_1, \cdots, e_T$), the LSTM computes a sequence of hidden states ($h_1, \cdots, h_T$). The final hidden state, or a pooled representation over all hidden states, is used as a global representation of the document. This representation is then fed into a fully connected layer and a softmax classifier. The LSTM is particularly suitable for modeling longer documents where topic cues may be distributed across distant parts of the text.

4.5. Bidirectional LSTM (BiLSTM)

The BiLSTM [6] extends the standard LSTM by employing two separate LSTMs: one processing the sequence from left to right and the other from right to left. The forward and backward hidden states are concatenated at each time step, yielding a richer contextual representation that incorporates both past and future context. The final document representation is obtained by pooling or concatenating the

terminal hidden states from both directions and passing the result to the classification layer. This bidirectional context modeling provides more informative representations, especially in cases where critical topic indicators depend on both preceding and following context.

4.6. Multilingual BERT (mBERT)

Multilingual BERT (mBERT) is a Transformer-based model [8] pretrained on multilingual Wikipedia data using masked language modeling and next sentence prediction objectives. In this study, mBERT is fine-tuned end-to-end for the Fulani topic classification task. Given an input sequence, mBERT produces contextualized token representations at each layer. The final hidden representation of the special [CLS] token is used as a sentence-level embedding and is passed to a task-specific linear layer followed by a softmax function to predict the topic class.

4.7. XLM-RoBERTa (XLM-R)

In addition to mBERT, this study evaluates XLM-R (XLM-RoBERTa) [10], a multilingual Transformer pretrained on a substantially larger and more diverse multilingual corpus. XLM-R has demonstrated strong cross-lingual transfer performance, particularly for low-resource languages, and is fine-tuned end-to-end in the same way as mBERT. By comparing mBERT and XLM-R, the study analyzes the impact of the scale and diversity of pretraining data on Fulani topic classification performance and assesses the robustness of Transformer-based transfer learning in this low-resource setting.

4.8. Training Objective and Optimization

All models are trained under a supervised multi-class classification setting using the categorical cross-entropy loss function:

$$L = -\left(y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + \cdots + y_C \log \hat{y}_C\right) = -\sum y_c \log \hat{y}_c \left(c = 1, \cdots, C\right)$$

where C is the number of topic classes, y_c denotes the ground-truth label for class c, and $\hat{y}_c$ represents the predicted probability for class c. To account for the moderate class imbalance (**Table 6**), class weights inversely proportional to the training-set class frequencies are included in the loss for all models. Optimization uses the Adam optimizer [22] (AdamW with weight decay 0.01 and 6% learning-rate warm-up for the Transformer-based models). To improve generalization and reduce overfitting, dropout [23] is applied to the embeddings and before the classifier, and early stopping is based on validation loss. Hyperparameters were selected by grid search on the validation set.

5. Experimental Setup

5.1. Data Splits and Implementation Details

The experiments were designed to ensure fair comparison between classical neural models and Transformer-based architectures under identical training condi-

tions. The constructed Fulani corpus was divided into three subsets: training, validation, and test sets. The training set contains 70% of the data and is used to train the classification models. The validation set contains 15% of the data and is used for hyperparameter tuning and early stopping. The remaining 15% (795 documents) constitutes the test set and is used exclusively for final performance evaluation. To ensure balanced representation of all topic categories across the splits, a stratified sampling strategy was employed, so that each class maintains approximately the same proportion in the training, validation, and test partitions.

All experiments were implemented in Python 3.10. Neural network models were developed using PyTorch 2.0 [24], the HuggingFace Transformers library (v4.28) [25] was used for fine-tuning mBERT and XLM-R, and metrics were computed with scikit-learn. Training was performed on a single workstation with an NVIDIA RTX 3090 GPU (24 GB). Embedding dimensions were 300 for the neural models and 768 for the Transformer models. Training stopped when the validation loss had not improved for five consecutive epochs or after at most 20 epochs, and the best checkpoint was restored. The test set was evaluated exactly once with the selected model. **Table 8** summarizes the key hyperparameters.

Table 8. Main hyperparameters used in the experiments.

Parameter	Value
Batch size	32
Learning rate	1×10^{-3} (neural models); 2×10^{-5} (Transformer models)
Embedding dimension	300 (neural models); 768 (Transformer models)
Dropout	0.3 on embeddings and 0.5 before the classifier (neural models); 0.3 on the [CLS] representation (Transformer models)
Maximum sequence length	256 tokens
Optimizer	Adam (neural models); AdamW, weight decay 0.01, 6% warm-up (Transformer models)
Maximum epochs	20
Early stopping	Validation loss, patience of 5 epochs
Random seeds	42, 123, 456

5.2. Evaluation Metrics

Standard multi-class classification metrics were employed: accuracy, precision, recall, and F1-score. Accuracy is the proportion of correctly classified documents. Precision, recall, and F1-score are computed per class and then macro-averaged (unweighted mean over the five classes); macro-averaging was chosen because the moderate class imbalance would otherwise let a model score well by performing well on the larger classes. To account for variability due to random initialization

and data ordering, every model was trained and evaluated with three random seeds (42, 123, 456) on the same stratified data splits, and we report the mean ± standard deviation (SD). The statistical significance of accuracy differences between adjacent models was assessed with paired bootstrap resampling (10,000 resamples of the test set).

6. Experimental Results

6.1. Overall Model Performance

Table 9 summarizes the classification performance of all evaluated models, and Figure 2 and Figure 4 visualize the comparison. Transformer-based models consistently outperform the classical neural architectures. Among the evaluated models, XLM-R achieved the highest performance across all evaluation metrics, improving accuracy by 1.7 points over mBERT, 4.9 points over BiLSTM, and 8.8 points over CNN. Paired bootstrap tests show that the accuracy differences between all adjacent models are statistically significant ($p < 0.01$); the narrowest margin, between mBERT and XLM-R, has a 95% confidence interval of [0.9, 2.5] percentage points.

Table 9. Performance comparison of classification models (%, mean ± SD over 3 seeds).

Model	Accuracy	Precision	Recall	F1-score
CNN	82.4 ± 0.6	81.7 ± 0.7	81.9 ± 0.5	81.8 ± 0.6
LSTM	84.1 ± 0.5	83.6 ± 0.6	83.5 ± 0.5	83.5 ± 0.5
BiLSTM	86.3 ± 0.4	85.7 ± 0.5	85.9 ± 0.4	85.8 ± 0.4
mBERT	89.5 ± 0.4	89.1 ± 0.4	89.0 ± 0.3	89.0 ± 0.4
XLM-R	**91.2 ± 0.3**	**90.8 ± 0.3**	**90.7 ± 0.3**	**90.7 ± 0.3**

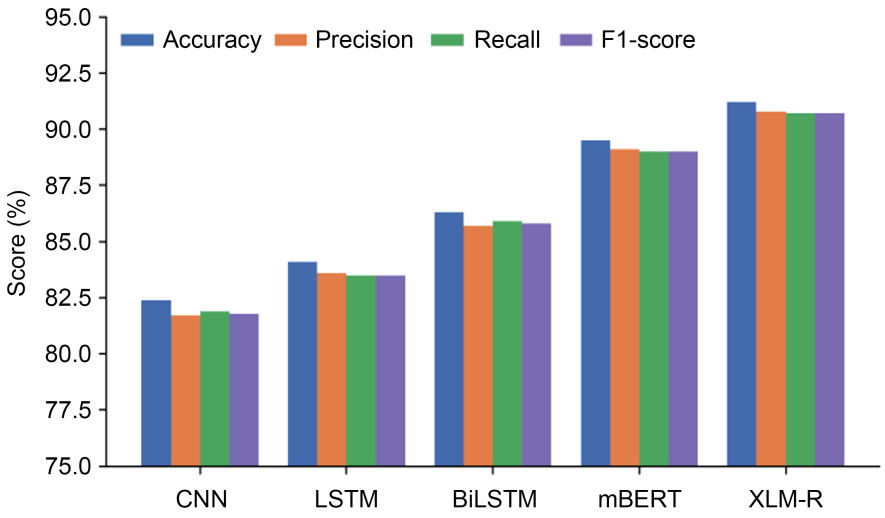


Figure 2. Performance comparison of classification models on the Fulani dataset.

6.2. Analysis of Neural Architectures

Among the neural models, BiLSTM achieved the best results. This improvement can be attributed to the bidirectional modeling of sequential information, which allows the model to capture contextual dependencies from both preceding and succeeding tokens. In contrast, CNN models primarily capture local n-gram patterns and therefore exhibit slightly lower performance when long-range dependencies are important.

6.3. Performance of Transformer Models

Transformer-based models demonstrated superior classification performance due to their ability to capture contextual semantic relationships across entire sequences. The self-attention mechanism enables the model to dynamically weigh token interactions regardless of their position within the text. The superior performance of XLM-R compared to mBERT can be explained by its larger pretraining dataset and improved training methodology, which enhance cross-lingual transfer capabilities for low-resource languages. Figure 3 shows the training and validation accuracy during fine-tuning, and Figure 4 compares the accuracy of all models.

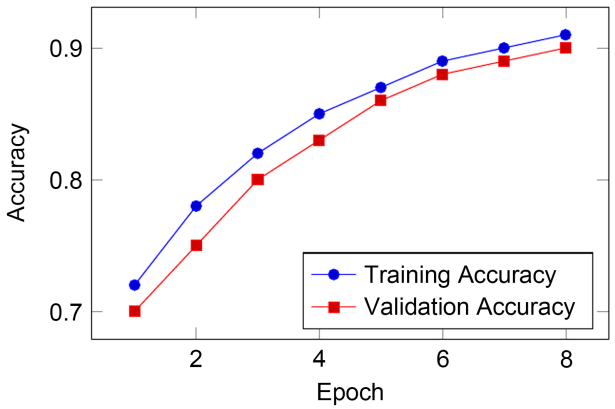


Figure 3. Training and validation accuracy during model training.

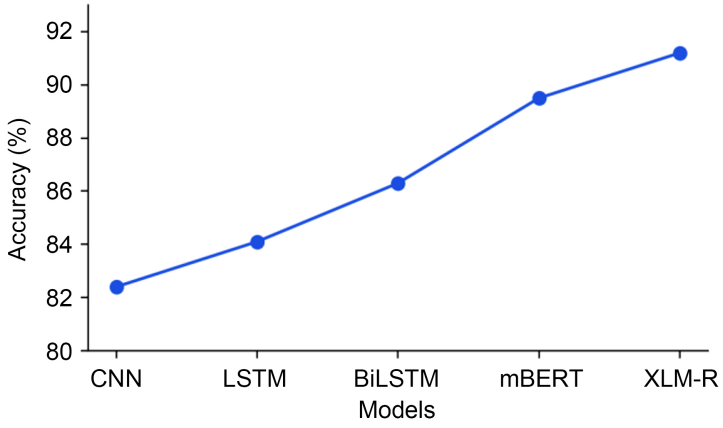


Figure 4. Accuracy comparison of evaluated models.

6.4. Error Analysis

Table 10 and Figure 5 show the per-class results and the confusion matrix of the best-performing model (XLM-R) on the test set. Politics (F1 93.9%) and Religion (92.7%) are the easiest classes, followed by Education (91.9%) and Health (91.2%). Culture is the hardest (84.0%). Of the 73 misclassified test documents, 25 are Culture documents, and their errors are spread over all four other classes (7 predicted as Religion, 7 as Education, 6 as Health, and 5 as Politics). The most confusable pairs therefore all involve Culture: culture-religion (13 errors in both directions), culture-education (12), and culture-health (11).

Table 10. Per-class results of XLM-R on the test set (%).

Class	Precision	Recall	F1-score	Support
Politics	93.8	94.1	93.9	180
Health	91.0	91.3	91.2	150
Education	91.5	92.3	91.9	165
Religion	92.4	93.1	92.7	158
Culture	85.3	82.7	84.0	142
Macro average	**90.8**	**90.7**	**90.7**	**795**

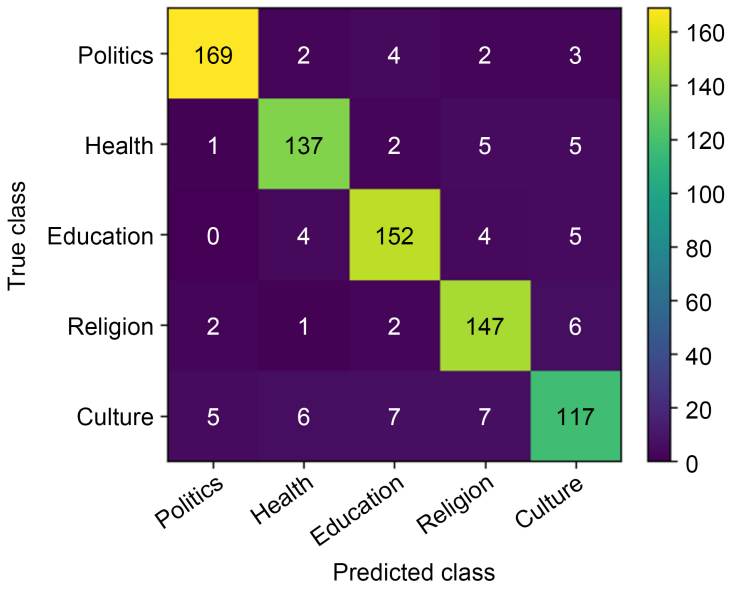


Figure 5. Confusion matrix of the best-performing model (XLM-R) on the 795-document test set. Diagonal cells are correctly classified documents; off-diagonal cells are misclassifications between topics.

Culture acts as a broad, umbrella category covering folklore, traditions, cuisine, and social practice, so it shares vocabulary with Religion (ritual and belief), Education (didactic and historical narrative), and Politics (customary governance).

Manual inspection of 40 misclassified Culture documents found that 22 (55%) contained substantial secondary content from another category, for example, a traditional naming ceremony that also described the accompanying religious blessing in detail. Part of the error therefore reflects genuine topical overlap rather than model failure, and Culture is also the category with the most annotator disagreement (Section 3.2). A second error source is short text: XLM-R accuracy is 84.7% on documents under 40 tokens versus 92.6% on documents at or above the median length of 116 tokens. Dialectal variation may also contribute, but its effect cannot be isolated because no per-variety evaluation was performed (Section 7.2).

7. Discussion and Limitations

7.1. Discussion

The results show a clear ordering of model families: pretrained multilingual Transformers outperform the BiLSTM, which in turn outperforms the unidirectional LSTM and the CNN. This suggests that, even for a language that is under-represented in pretraining data, transfer from multilingual pretraining is more effective than training task-specific encoders on a corpus of about 5000 documents. The larger and more diverse pretraining corpus of XLM-R gives it a further advantage over mBERT. The error analysis indicates that remaining errors are concentrated in the Culture category, which overlaps topically with the other classes, rather than being spread evenly.

7.2. Limitations

Several limitations should be considered when interpreting the results. *Source bias.* All texts were collected from online and broadcast sources (news portals, educational and government websites, religious and cultural blogs, Facebook groups, and radio transcripts). The corpus therefore over-represents written, edited, and internet-accessible Fulani, skews toward religious and some educational content, and under-represents oral traditions and communities with limited online presence. *Exclusion of code-mixed documents.* Documents with heavy code-mixing with French, English, or Arabic were excluded to preserve linguistic integrity. Because code-mixing is common in everyday Fulani writing, performance on unfiltered real-world text is likely lower than reported here. *Generalization across varieties.* Fula is a dialect continuum, but the corpus is treated as a single dataset and was not formally audited for dialectal coverage. Western varieties (Pulaar and Pular) are over-represented, so the results reflect the corpus's particular dialectal mixture and may not generalize equally to eastern varieties. Only Latin-script text is covered; Ajami and Adlam scripts are out of scope. *Pretraining coverage.* To the best of our knowledge, neither mBERT nor XLM-R includes Fula among its pretraining languages, so their performance relies on cross-lingual transfer. *Scale and task scope.* The corpus is small (5300 documents, 3710 for training), and the task uses five coarse, single-label topics; multi-topic documents and finer-grained taxonomies were not considered. Future work should address these limitations with

larger, more varied, and variety-labeled corpora.

8. Conclusions and Future Work

This study presented one of the first systematic investigations of automatic topic classification for the Fulani language. Due to the lack of existing datasets and computational resources for Fulani, a new labeled corpus was constructed and manually annotated across five topical categories: politics, health, education, religion, and culture. A comprehensive experimental framework was developed to evaluate multiple deep learning approaches, including CNN, LSTM, BiLSTM, and Transformer-based models such as mBERT and XLM-R. The results demonstrate that while neural architectures achieve competitive performance, Transformer-based models consistently outperform them in terms of classification accuracy and F1-score. The findings highlight the effectiveness of pretrained multilingual language models for low-resource languages and emphasize the importance of dataset construction and linguistic preprocessing in achieving reliable NLP performance.

Future work will focus on several directions. First, expanding the Fulani dataset with larger and more diverse sources, including code-mixed text and balanced coverage of Fula varieties, will improve model robustness and generalization. Second, domain adaptation and cross-lingual transfer techniques may further enhance performance by leveraging related languages within the Niger-Congo family. Third, future research may explore additional NLP tasks for Fulani, including sentiment analysis, named entity recognition, and machine translation. By providing both a benchmark dataset and baseline experimental results, this study lays the foundation for future computational research on Fulani and contributes to the broader effort of promoting linguistic diversity in natural language processing.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Joshi, P., Santy, S., Budhiraja, A., Bali, K. and Choudhury, M. (2020) The State and Fate of Linguistic Diversity and Inclusion in the NLP World. *Proceedings of the* 58*th Annual Meeting of the Association for Computational Linguistics*, Online, 5-10 July 2020, 6282-6293. https://doi.org/10.18653/v1/2020.acl-main.560

[2] Hedderich, M.A., Lange, L., Adel, H., Strötgen, J. and Klakow, D. (2021) A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. *Proceedings of the* 2021 *Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, Online, 6-11 June 2021, 2545-2568. https://doi.org/10.18653/v1/2021.naacl-main.201

[3] Kim, Y. (2014) Convolutional Neural Networks for Sentence Classification. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (EMNLP), Doha, 25-29 October 2014, 1746-1751. https://doi.org/10.3115/v1/d14-1181

[4] Zhang, X., Zhao, J. and LeCun, Y. (2015) Character-Level Convolutional Networks for Text Classification. *Advances in Neural Information Processing Systems*, **28**, 649-657.

[5] Hochreiter, S. and Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Computation*, **9**, 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[6] Schuster, M. and Paliwal, K.K. (1997) Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, **45**, 2673-2681. https://doi.org/10.1109/78.650093

[7] Vaswani, A., Shazeer, N., Parmar, N., *et al.*, (2017) Attention Is All You Need. *Advances in Neural Information Processing Systems*, **30**, 5998-6008.

[8] Devlin, J., Chang, M., Lee, K. and Toutanova, K. (2019) BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *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), Minneapolis, 2 June-7 June 2019, 4171-4186. https://doi.org/10.18653/v1/n19-1423

[9] Ogueji, K., Zhu, Y. and Lin, J. (2021) Small Data? No Problem! Exploring the Viability of Pretrained Multilingual Language Models for Low-Resourced Languages. *Proceedings of the* 1*st Workshop on Multilingual Representation Learning*, Punta Cana, 11 November 2021, 116-126. https://doi.org/10.18653/v1/2021.mrl-1.11

[10] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., *et al.* (2020) Unsupervised Cross-Lingual Representation Learning at Scale. *Proceedings of the* 58*th Annual Meeting of the Association for Computational Linguistics*, Online, 5-10 July 2020, 8440-8451. https://doi.org/10.18653/v1/2020.acl-main.747

[11] Lample, G. and Conneau, A. (2019) Cross-Lingual Language Model Pretraining. *Advances in Neural Information Processing Systems*, **32**, 7059-7069.

[12] Adebara, I. and Abdul-Mageed, M. (2022) Towards Afrocentric NLP for African Languages: Where We Are and Where We Can Go. *Proceedings of the* 60*th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), Dublin, 22-27 May 2022, 3814-3841. https://doi.org/10.18653/v1/2022.acl-long.265

[13] Adelani, D.I. Abbott, J., Neubig, G., D'souza, D., Kreutzer, J., Lignos, C., *et al.*, (2021) MasakhaNER: Named Entity Recognition for African Languages. *Transactions of the Association for Computational Linguistics*, **9**, 1116-1131.

[14] Muhammad, S.H., Abdulmumin, I., Yimam, S.M., Adelani, D.I., Ahmad, I.S., Ousidhoum, N., *et al.* (2023) Semeval-2023 Task 12: Sentiment Analysis for African Languages (Afrisenti-Semeval). *Proceedings of the* 17*th International Workshop on Semantic Evaluation* (SemEval-2023), Toronto, 13-14 July 2023, 2319-2337. https://doi.org/10.18653/v1/2023.semeval-1.315

[15] Alabi, J.O., Adelani, D.I., Mosbach, M. and Klakow, D. (2022) Adapting Pretrained Language Models to African Languages via Multilingual Adaptive Finetuning. *Proceedings of the* 29*th International Conference on Computational Linguistics*, Gyeongju, 12-17 October 2022, 4336-4349.

[16] Dossou, B.F.P., Tonja, A.L., Yousuf, O., Osei, S., Oppong, A., Shode, I., *et al.* (2022) AfroLM: A Self-Active Learning-Based Multilingual Pretrained Language Model for 23 African Languages. *Proceedings of the Third Workshop on Simple and Efficient Natural Language Processing* (SustaiNLP), Abu Dhabi, 7 December 2022, 52-64. https://doi.org/10.18653/v1/2022.sustainlp-1.11

[17] Adebara, I., Elmadany, A., Abdul-Mageed, M. and Alcoba Inciarte, A. (2023) SERENGETI: Massively Multilingual Language Models for Africa. *Findings of the Association for Computational Linguistics*: *ACL* 2023, Toronto, 9-14 July 2023, 1498-1537. https://doi.org/10.18653/v1/2023.findings-acl.97

[18] Adelani, D.I., Masiak, M., Azime, I.A., Alabi, J., *et al.* (2023) MasakhaNEWS: News Topic Classification for African Languages. *Proceedings of the* 13*th International Joint Conference on Natural Language Processing and the* 3*rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, Nusa Dua, 1-4 November 2023, 144-159.

[19] Costa-Jussà, M.R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., *et al.* (2024) Scaling Neural Machine Translation to 200 Languages. *Nature*, **630**, 841-846. https://doi.org/10.1038/s41586-024-07335-x

[20] Kudo, T. and Richardson, J. (2018) Sentencepiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. *Proceedings of the* 2018 *Conference on Empirical Methods in Natural Language Processing*: *System Demonstrations*, Brussels, 31 October-4 November 2018, 66-71. https://doi.org/10.18653/v1/d18-2012

[21] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2017) Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, **5**, 135-146. https://doi.org/10.1162/tacl_a_00051

[22] Kingma, D.P. and Ba, J. (2015) Adam: A Method for Stochastic Optimization. 3*rd International Conference on Learning Representations* (*ICLR* 2015), San Diego, 7-9 May 2015.

[23] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, **15**, 1929-1958.

[24] Paszke, A., Gross, S., Massa, F., Lerer, A., *et al.* (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, **32**, 8024-8035.

[25] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., *et al.* (2020) Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the* 2020 *Conference on Empirical Methods in Natural Language Processing*: *System Demonstrations*, Online, 16-20 November 2020, 38-45. https://doi.org/10.18653/v1/2020.emnlp-demos.6