MultiProp Framework: Ensemble Models for Enhanced Cross-Lingual Propaganda Detection in Social Media and News using Data Augmentation, Text Segmentation, and Meta-Learning

Farizeh Aldabbas[*,†], **Shaina Ashraf** [*,†], **Rafet Sifa** [*,‡], **Lucie Flek**[*,†]

[*]University of Bonn, [†]Conversational AI and Social Analytics (CAISA) Lab, [‡]Fraunhofer IAIS

fi.db.73@gmail.com, sashraf@bit.uni-bonn.de
Rafet.Sifa@iais.fraunhofer.de, flek@bit.uni-bonn.de

Abstract

Propaganda, a pervasive tool for influencing public opinion, demands robust automated detection systems, particularly for under-resourced languages. Current efforts largely focus on well-resourced languages like English, leaving significant gaps in languages such as Arabic. This research addresses these gaps by introducing MultiProp Framework, a cross-lingual meta-learning framework designed to enhance propaganda detection across multiple languages, including Arabic, German, Italian, French and English. We constructed a multilingual dataset using data translation techniques, beginning with Arabic data from PTC and WANLP shared tasks, and expanded it with translations into German Italian and French, further enriched by the SemEval23 dataset. Our proposed framework encompasses three distinct models: MultiProp-Baseline, which combines ensembles of pre-trained models such as GPT-2, mBART, and XLM-RoBERTa; MultiProp-ML, designed to handle languages with minimal or no training data by utilizing advanced meta-learning techniques; and MultiProp-Chunk, which overcomes the challenges of processing longer texts that exceed the token limits of pre-trained models. Together, they deliver superior performance compared to state-of-the-art methods, representing a significant advancement in the field of cross-lingual propaganda detection.

1 Introduction

Propaganda detection in text has gained significant attention, driven by the need to identify biased or misleading content across various platforms. While progress has been made, research remains predominantly focused on English, leaving other languages, especially those with fewer resources, under-explored. The lack of annotated datasets in these languages poses a significant challenge to developing effective detection systems.

To address this challenge, various data augmentation techniques, such as oversampling (Chavan and Kane, 2022), and data translation (Amihaesei et al., 2023), have been explored.

However, annotated resources for low-resource languages remain a significant challenge, emphasizing the need for more comprehensive frameworks.

In addition, studies have shown that while data augmentation can boost performance, an excess can lead to issues like label loss in translated texts. This underscores the need for models capable of learning from limited samples or adapting to new tasks with minimal training, paving the way for zero-shot and few-shot learning approaches. Our contributions to this field include:

1. MultiProp Dataset: We introduce MultiProp, a combined dataset that integrates data from the PTC dataset (Martino et al., 2020), SemEval 2023 (Piskorski et al., 2023) , and WANLP (Mittal and Nakov, 2022), resulting in a robust multilingual dataset that includes Arabic, addressing the data scarcity in low-resource languages.

2. MultiProp-Baseline: Our base model allows for flexibility in choosing between three ensemble architectures, combining transformer-based models, GloVe (Pennington et al., 2014) embeddings, and FastText (Bojanowski et al., 2017) to harness their collective strengths.

3. MultiProp-Chunk: To overcome the limitations of pre-trained models with long texts, we developed MultiProp-Chunk, which segments text into chunks, preserving textual continuity across segments.

4. MultiProp-ML(MetaLearner): Our model employs few-shot and zero-shot learning across seven languages, consistently outperforming strong ensemble baselines, including Multilingual BERT [1], XLM-RoBERTa [2] and GPT2 [3] as well as monolin-

[1] https://huggingface.co/google-bert/bert-base-multilingual-uncased

[2] https://huggingface.co/FacebookAI/xlm-roberta-large

[3] https://huggingface.co/openai-community/

Processings of the 1st Workshop on NLP for Languages Using Arabic Script (AbjadNLP), pages 7–22
January 19, 2025.

gual models trained on Arabic [4].

5. Ensemble Models: We investigated various ensembling strategies, combining the strengths of multiple pre-trained models across encoder-based, decoder-based, and hybrid architectures, as well as both multilingual and monolingual models. For final predictions, we utilized an additional ensemble of machine learning classifiers, including SVM (Chang and Lin, 2011) and Random Forest (Breiman, 2001), to enhance performance across diverse linguistic contexts.

2 Related Work

Propaganda detection in text has garnered significant attention due to the need to identify biased or misleading content. Despite advancements in English, low-resource languages lack annotated datasets, limiting detection system development.

Early efforts, such as (Barrón-Cedeno et al., 2019), used binary classification (propaganda vs. non-propaganda), while (Habernal et al., 2017) annotated a corpus with five fallacies tied to propaganda techniques. NLP4IF-2019 (Da San Martino et al., 2019) marked a milestone by curating a dataset of 18 persuasive techniques within English news articles, forming a foundation for further research.

Recent advancements, including SemEval23 (Piskorski et al., 2023), have extended propaganda detection to multilingual contexts. In contrast, research involving Arabic has been sparse, with notable exceptions such as the WANLP 2022 Shared Task for Arabic propaganda detection (Mittal and Nakov, 2022). However, the data provided for Arabic in these tasks has been limited and suffers from imbalanced labels, which magnifies the challenge of training effective models.

Cross-lingual transfer and data-efficient models offer promising solutions by leveraging knowledge from resource-rich languages. Data augmentation methods, such as (back)translation, play a crucial role in cross-lingual propaganda detection, enabling the creation of additional samples to expand datasets for low-resource languages (Hromadka et al., 2023; Falk et al., 2023).

Building upon these methods, recent advancements in the field have focused on leveraging cross-lingual transfer learning and meta-learning approaches. For example, LaBSE (Feng et al., 2020) enhances performance for low-resource languages by integrating pre-training with dual-encoder fine-tuning. Researchers like (Brown et al., 2020) and (Lauscher et al., 2020) have addressed the challenges of domain shifts across languages, highlighting the effectiveness of few-shot and zero-shot learning techniques to minimize dependence on extensive annotated data.

Additionally, (Nooralahzadeh et al., 2020) introduced cross-lingual meta-learning architectures designed to optimize learning with minimal training instances.

The field has also seen the adoption of ensemble learning techniques to boost model performance. Methods such as boosting, exemplified by AdaBoost (Freund et al., 1996), and bagging approaches (Breiman, 1996) like random forests (Breiman, 2001), combine multiple models to enhance classification accuracy. Voting methods, both hard and soft (Kandasamy et al., 2021), aggregate predictions from various classifiers to achieve better performance. Stacking, as described by (Ting and Witten, 1997), employs a meta-learner to integrate outputs from base models, thereby improving robustness and generalization.

A significant challenge remains the 512-token limit of pre-trained transformer models like BERT, which can lead to the loss of essential contextual information when longer documents are truncated (Xie et al., 2020). Although Longformer (Beltagy et al., 2020) mitigates this issue with a global attention mechanism to handle longer texts, it often requires task-specific adjustments that are not universally applicable. Inspired by the approach in (Pappagari et al., 2019), which splits text into fixed-size overlapping segments and uses BERT to extract segment-level representations, followed by an LSTM layer (Hochreiter, 1997) or small transformer model to generate document-level embeddings,We developed a similar approach by replacing the LSTM and transformer with an attention layer to process segment-level embeddings. Additionally, we maintained the use of overlapping segments to ensure context is preserved across chunks. By leveraging ensemble methods and cross-lingual transfer learning, our work seeks to improve model adaptability and accuracy across diverse languages and text lengths.

3 MultiProp Data

The primary goal of our study was to address the shortage of Arabic propaganda detection datasets.

gpt2-large

[4] https://huggingface.co/aubmindlab/aragpt2-mega-detector-long

Building on this foundation, we expanded our research to develop a cross-lingual propaganda detection framework that includes Arabic, a language often underrepresented in previous studies on persuasion techniques.To achieve this,we combined datasets from various shared tasks to create the MultiProp dataset, a multilingual resource supporting diverse languages, which includes 18 final labels corresponding to different propaganda techniques. Table 1 provides statistics for MultiProp and its sources. The MultiProp dataset includes:

Arabic:This dataset was sourced from the WANLP22 shared task and augmented with translated PTC-SemEval20 data. Preprocessing steps included standardizing labels, replacing links with 'URL' and '@name' with 'USR', and filtering out instances that lacked any techniques (labeled as 'no technique').

***German*:** This dataset is compiled from the SemEval2023 shared task dataset and translated PTC-SemEval20 data. To align the labels with other datasets, redundant techniques were removed, and instances without any remaining techniques were discarded. The test data comprises translated PTC English data.

***English*:** Derived from SemEval23 data and supplemented with German data translated into English. Preprocessing included URL removal and standardization.

***French*:** Sourced from SemEval23 and harmonized with the translated PTC data.

***Italian*:** Also drawn from SemEval23 and aligned with the translated PTC data.

Polish and Russian: The development sets from SemEval23 were included and used as test sets, as the SemEval23 test sets are not accessible. This allows for evaluating model performance on "surprise" languages that were not seen during training. For detailed statistics on the MultiProp dataset and the number of instances for each language, refer to Table 4 in the appendix.

Table 1: Dataset statistics for MultiProp and its sources.

Dataset	Train	Dev	Test	Num Classes	Source
WANLP (ar)	504	52	52	21	Tweets
PTC(en)	293	57	101	18	News Articles
SemEval23(en)	446	90	54	23	News Articles
SemEval23(de)	132	45	50	23	News Articles
MultiProp (ar)	517	68	68	18	Tweets & Articles
MultiProp (en)	488	143	101	18	News Articles

4 Methodology

The MultiProp Framework, depicted in Figure 1, comprises three variants: MultiProp-Baseline, MultiProp-ML, and MultiProp-Chunk.While MultiProp-Baseline maintains a consistent core architecture, MultiProp-ML and MultiProp-Chunk introduce additional steps to address specific challenges. Our approach integrates GloVe and FastText embeddings (GloFast) with transformer models to build three ensemble architectures: encoder-based, decoder-based, and hybrid, utilizing Use-FFN and Skip-FFN methods for final predictions.

The systems were evaluated in two settings: zero-shot, where models were trained exclusively on English and German data, with Arabic as the target language and French, Italian, Polish, and Russian included in the evaluation to assess their ability to generalize across diverse languages; and few-shot, where models were trained on extensive English data and a limited number of instances (5-shot, 4 ways) from Arabic, German, French, and Italian datasets, with Polish and Russian included as surprise languages in the testing phase. We will now discuss the three developed systems in detail:

4.1 MultiProp-Baseline

The MultiProp-Baseline model features two key components: embeddings generation and predictions aggregation. In the embeddings generation phase, textual content is converted into numerical representations through various embedding techniques. The predictions aggregation phase then combines these representations using multiple ensemble methods to produce the final predictions.

4.1.1 Embeddings Generation

We explore a variety of embedding techniques, from traditional methods like TF-IDF to advanced approaches such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2017), and transformer-based models (Vaswani et al., 2017). Our novel approach integrates these baseline techniques with transformer-based embeddings to rich, nuanced representations that combine different levels of semantic information, strengthening its capacity to understand and process the input data across different languages.

a) GloFast Embedding: For generating word embeddings, we combined GloVe and FastText models, training them on the MultiProp dataset, which encompasses English, Arabic, and German texts. The preprocessing steps involved lowercasing, retaining stop words, removing punctuation,

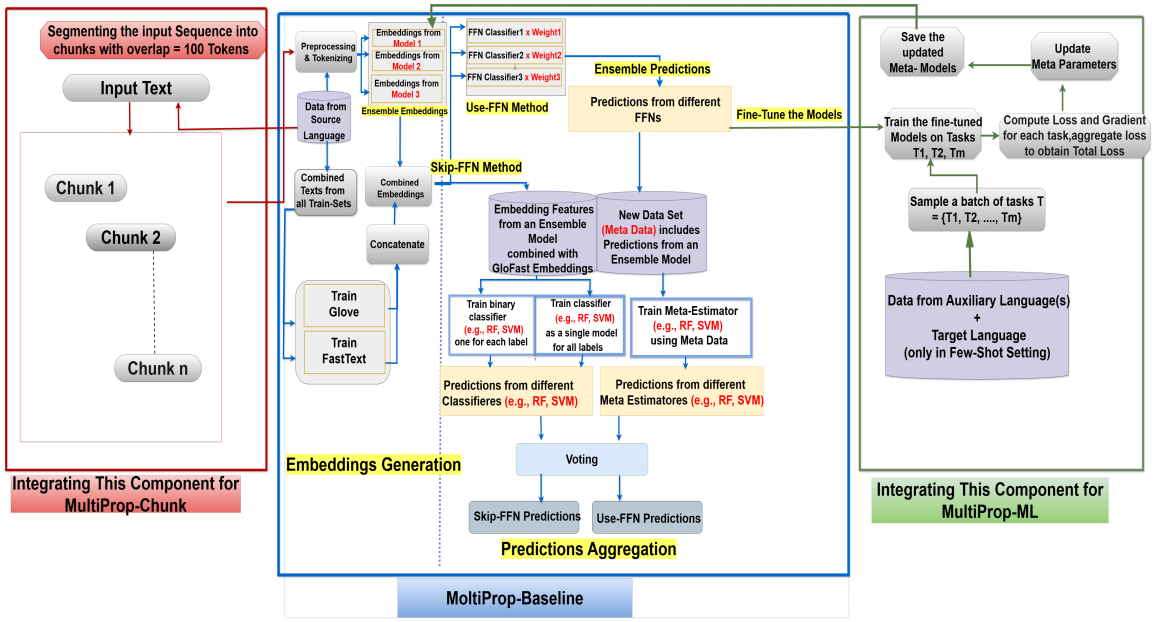

Figure 1: An Overview of MultiProp Framework

and handling out-of-vocabulary (OOV) words. The embeddings from both models were concatenated to form unified vectors, either 600 dimensions (300 from GloVe and 300 from FastText) or 200 dimensions (100 from each), and were integrated with transformer-based embeddings to improve pattern recognition in the text.

b) Transformer-Based Embedding: Transformer models, such as BERT (Devlin, 2018) and GPT (Radford et al., 2019), utilize self-attention mechanisms to capture both global and local word dependencies. Drawing inspiration from prior research that highlights the benefits of combining diverse embedding methods (Sifa et al., 2019), (Heinisch et al., 2023), our approach integrates transformer-based models with GloFast embeddings. To enhance our classifiers' ability to capture complex patterns in text, we employed three types of ensembles: encoder-based, decoder-based, and hybrid architectures.

1. The encoder-based ensemble model integrates multilingual transformers like mBERT (Devlin, 2018) and XLM-RoBERTa (Conneau et al., 2019), along with monolingual models such as AraBERT (Antoun et al., 2020). Pretrained on masked language modeling tasks across up to 104 languages, these models excel in cross-lingual transfer learning.

2. The decoder-based ensemble model utilizes GPT variants, including GPT-2 medium, GPT-2 large, and AraGPT2 (Radford et al., 2019). While GPT-2 models are primarily pretrained on English, AraGPT2 extends this to Arabic, and these models leverage decoder architectures for text generation. They are well-suited for generating coherent text, summarization, and translation tasks, with their multilingual capabilities enhancing their overall performance.

3. The encoder-decoder-based ensemble model (hybrid) combines models like mBART50 (Tang et al., 2020) and mT5 (Xue et al., 2020), pretrained on sequence-to-sequence tasks across up to 101 languages. This hybrid approach merges the strengths of both encoder and decoder architectures, making it highly effective for translation, summarization, and text generation. AraBART (Eddine et al., 2022) is also included for enhanced support of Arabic.

4.1.2 Predictions Aggreagation

The combined embeddings are fed into classifiers or meta-estimators. These classifiers include traditional machine learning algorithms such as Support Vector Machines (SVM) (Chang and Lin, 2011), Logistic Regression (LR) (Cox, 1959), and Random Forest (RF) (Breiman, 2001). Alternatively, these machine learning models function as meta-estimators when trained on the predictions generated by base classifiers (also known as level-0 classifiers), such as the Feed-Forward Neural Network (FFN) in our approach, further refining and

improving final prediction accuracy. For prediction aggregation, we employ two key methods:

a) Use-FFN Method In this method, the combined embeddings are first passed through a fully connected neural network (FFN) with three linear layers and two ReLU activation functions, serving as the base learner. Adopting a stacking approach, predictions from various transformer-based models (PLMs), each paired with an FFN, are aggregated to form a new dataset. To formalize, let E_{i,PLM_j} denote the embedding of the i-th instance produced by the j-th transformer model in the ensemble. The final embedding for the i-th instance is computed as follows. For each model j, concatenate the model's embeddings with GloVe and FastText embeddings:

$$E_{i,\text{final}_j} = E_{i,\text{PLM}_j} \oplus E_{i,\text{GloVe}} \oplus E_{i,\text{FastText}}$$

The concatenated embeddings E_{i,final_j} for each model j are then passed through their respective feed-forward networks (FFNs). Each FFN outputs logits, which are then transformed into prediction probabilities for each label by applying a sigmoid activation function:

$$\hat{y}_{i,j} = \sigma(P_{\text{FFN}_j})$$

where P_{FFN_j} represents the logits output by the FFN of the j-th model.

A threshold of 0.5 is applied to each label's prediction to select the most confident predictions, ensuring stable and accurate outputs for creating the new dataset. This dataset, containing the gold labels, combines predictions from different models within the ensemble.

Finally, predictions from multiple level-1 classifiers (meta-estimators) are aggregated for each label using majority voting. Let $\hat{y}_i^{(m)}$ represent the prediction from the m-th classifier for the i-th input text. The final prediction is determined as:

$$\hat{y}_i^{\text{final}} = I\left(\frac{1}{M}\sum_{m=1}^{M}\hat{y}_i^{(m)} \geq \frac{1}{2}\right) \quad (1)$$

where M is the number of classifiers, and $I(\cdot)$ is the indicator function that outputs 1 if the condition is true and 0 otherwise.

This ensemble incorporates various classifiers, including Support Vector Machine (SVM) (Chang and Lin, 2011), Logistic Regression (LR) (Cox, 1959), Random Forest (RF) (Breiman, 2001), Gaussian Naive Bayes (Jahromi and Taheri, 2017) and XGBoost . Additionally, a hard voting approach is employed to aggregate the outputs of the meta-estimators. This ensemble method proved effective in detecting propagandistic techniques in news articles and tweets, resulting in a significant improvement in the overall F1 score by over 13%, demonstrating its robustness in multi-label classification tasks.

b) Skip-FFN Method The Skip-FFN method leverages embedding features to train multiple machine learning models that act as classifiers. This approach can be implemented in two ways: either by training each classifier to recognize patterns across all classes using non-linear kernels or by training each model as a binary classifier, focusing on individual classes. In this ensemble method, embeddings from various models, including GloFast embeddings, are concatenated for each classifier. This approach supports both monolingual and multilingual models, corresponding to three ensemble architectures: encoder-based, decoder-based, and hybrid models. The diversity of these models enhances the ability of classifiers to identify patterns across languages, improving the classification of text into different propaganda techniques.

The selection of models for multilingual propaganda detection is guided by recent research (Hromadka et al., 2023) and depends on the research objectives, target languages, and dataset characteristics. In our approach, we combined multilingual models with Arabic monolingual models to enhance performance in Arabic while ensuring consistent accuracy across all languages and avoiding bias toward any particular language.

The predictions from the classifiers, whether trained on all classes or as binary classifiers, are denoted as $P_{\text{SVM}}, P_{\text{LR}}, P_{\text{RF}}, P_{\text{XGB}}, P_{\text{Gau}}$, corresponding to the outputs of Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), XGBoost (XGB), and Gaussian Naive Bayes (Gau) models, respectively. These predictions are processed and aggregated for each label. A threshold of 0.35 is applied to each prediction:

$$\hat{y}_i^{(m)} = I(P_i^{(m)} \geq 0.35)$$

where $P_i^{(m)}$ represents the prediction probability for the i-th instance from the m-th classifier.

After applying the threshold, the final prediction is generated through majority voting, as defined in Equation 1, which combines the predictions from all classifiers to enhance reliability.

To tackle the issue of imbalanced label distribution, which is typical in multi-label classification tasks, class weights are adjusted using the Class Weights Based on Frequency (CWBF) approach (Kim and Bethard, 2020). The weight for each class i is computed as follows:

$$w_i = \frac{f_{\text{max}}}{f_i}$$

where f_i is the frequency of class i in the training data, and f_{max} is the frequency of the most common class. This weighting scheme ensures that less frequent classes are given higher importance, reducing the likelihood of misclassification for these underrepresented classes. The computed weights are then applied during training with the Binary Cross-Entropy Loss with Logits, which is used as the loss function (see Appendix B.1 for further details).

4.2 MultiProp-ML

Meta-learning, often referred to as "learning-to-learn," focuses on creating models capable of quickly adapting to new tasks or domains with minimal labeled data, while avoiding overfitting (Nooralahzadeh et al., 2020). This adaptability is achieved by training the model during a meta-learning phase on a diverse set of tasks, equipping it to rapidly adjust to new tasks with only a few examples. Our approach employs a gradient-based meta-learning technique, explicitly optimizing the model for fast adaptation with minimal data, even in zero-shot settings where no labeled samples of the target language are available.

To this end, we present MultiProp-ML, a cross-lingual meta-learning model designed for adaptability. Ensemble models are initially pre-trained on English datasets to establish a robust linguistic foundation, then fine-tuned to effectively transition and adapt to low-resource languages.

During the meta-learning phase on auxiliary languages, the models are trained on batches of tasks, each derived from randomly sampled subsets of development data from auxiliary low-resource languages. For each task, a portion of the data (D_{train}) is used to update the model's parameters via gradient descent, and task-specific losses are computed based on this data. These losses are then summed across tasks to calculate a meta-loss, which is used to further update the model's parameters. In the few-shot learning stage, the models are evaluated on the target language (Arabic) using a labeled subset of the target language(D_{test}), after the meta-learner has been trained on labeled samples ((D_{train}) from the same language to simulate real-world conditions.

Alternatively, in the zero-shot setting, we utilize pseudo-labeling by generating pseudo-labels from high-confidence predictions (above a threshold of 0.6). These pseudo-labels are iteratively used to refine the model's performance, following the approach of (Awal et al., 2023).

4.2.1 MultiProp-ML Algorithm

As shown in Algorithm 4.2.1, each model in the ensemble is fine-tuned on English to initialize its parameters. To enhance feature representation, external embeddings, such as GloFast, are concatenated with the model's native embeddings. In the few-shot approach, the model leverages a limited amount of labeled data from the target language. For zero-shot learning, the model is trained using meta-task data from auxiliary languages.

Algorithm: MultiProp-ML

Fine-tune models M_i on source language h and initialize parameters θ_i.
if S is zero-shot **then**
  Utilize meta-task data from h and auxiliary languages, and apply self-training using pseudo labels from tgt.
else
  Utilize few-shot data with limited labels from all languages in L, excluding surprise languages.
end if
while not converged **do**
  Sample tasks $T = \{T_1, \ldots, T_m\}$ from D.
  for all models M_i in ensemble **do**
    for all tasks $T_j \in T$ **do**
      Compute gradients $\nabla_{\theta_i} L_{T_j}(M_i)$ and update parameters θ_i'.
    end for
    Update meta-parameters θ_i with learning rate β.
  end for
end while
Save meta-trained models M_i and evaluate on target language tgt.

4.2.2 What makes our MultiProp-ML approach different?

Our approach enriches the meta-learner with external embeddings, such as GloFast (a combination of GloVe and FastText), to improve generalization in zero-shot settings. Additionally, we employ an ensemble of models to enhance robustness and leverage multi-task learning on external classification tasks, including Arabic sentiment detection and framing detection, to further boost the model's adaptability across diverse tasks and languages in both zero-shot and few-shot scenarios. This combination of techniques allows our model to better generalize across different languages and domains, making it highly effective for cross-lingual tasks such as propaganda detection .

4.3 MultiProp-Chunk

In our third approach, we tackle the issue of processing text sequences that exceed the standard 512-token limit of most pretrained models. This is crucial for handling lengthy articles, which often exceed 1,000 tokens in our dataset 2, and for multi-label classification where relevant labels may be dispersed throughout the text. Our method builds upon the MultiProp-Baseline but incorporates additional processing steps. Text is first chunked into 512-token segments with a 100-token overlap to preserve context. Each chunk is then tokenized and processed through the ensemble models to generate embeddings, which are concatenated with GloFast embeddings. To aggregate the concatenated embeddings from different segments, we use an attention layer. This layer consists of a linear layer and a softmax function. It generates attention weights for each segment, which are used to scale the embeddings, assigning greater importance to more relevant segments.The final embeddings are then used for classification. Predictions are obtained by applying either the Skip-FFN or Use-FFN methods and taking a majority vote from various meta-learners or classifiers.

5 Experimental Setup

Through extensive experimentation, we identified the optimal learning rates for each model in our ensemble. This was achieved by leveraging both prior research and our own empirical testing. The final learning rates, provided as a list with one value per model, follow established best practices (see Table 6 in the appendix). We found that a batch size of 10 was ideal, and improvements in loss metrics plateaued after 5 epochs. Tokenization length was set to 512 to balance context retention with memory constraints, and a dropout rate of 0.1 was applied to mitigate overfitting. We employed the AdamW optimizer across all models due to its proven effectiveness with transformer architectures and ability to handle sparse gradients. Key hyperparameters for each model are detailed in Table 7. For generating predictions, we utilized advanced classifiers and meta-estimators. These classifiers' parameters were optimized using grid search with cross-validation on the development sets, with the results summarized in the appendix (see Table 8). Our ensemble framework was implemented in Python 3.9, using the PyTorch library. To manage memory constraints, we limited the maximum number of chunks generated during the tokenization process to avoid overwhelming the device, which was an NVIDIA A100-SXM4-80GB.

6 Results and Analysis

The results in Table 3 highlight the performance of the MultiProp Framework across seven languages, using three ensemble architectures Encoder-Based, Decoder-Based, and Hybrid models with two aggregation methods: Use-FFN and Skip-FFN.

The MultiProp-Chunk Hybrid model excels in Arabic and Russian, effectively handling long texts and preserving context. This capability is particularly valuable for detecting subtle propaganda techniques like *Appeal to Fear/Prejudice*, *Red Herring*, *Black-and-White Fallacy/Dictatorship*, and *Exaggeration/Minimization*, which require nuanced contextual understanding and linguistic complexity.

The MultiProp-Baseline En-B model delivers consistent and balanced results, particularly in Polish and Italian, making it a reliable choice for achieving stable outcomes. The MultiProp-ML approach demonstrates strong cross-lingual adaptability, with significant improvements in Italian and French when using the En-B or Hybrid architecture with Skip-FFN. It also boosts performance in English (source), German (auxiliary), and Arabic (target) by leveraging effective meta-learning.

When examining the diverse ensemble architectures in Arabic, distinct patterns emerge. Encoder-Based models excel at detecting nuanced labels such as *Appeal to Fear/Prejudice*, likely due to their ability to capture fine-grained contextual dependencies. Decoder-Based models perform better for labels like *Causal Oversimplification*, potentially benefiting from their sequence-generating

MultiProp-Baseline							
Ensemble Models	**En**	**Ar**	**Ge**	**It**	**Fr**	**Po**	**Ru**
En-B (Use-FFN)	0.434	0.416	0.457	0.404	0.509	0.480	0.420
En-B (Skip-FFN)	0.556	**0.569**	0.573	**0.573**	0.559	**0.605**	0.539
De-B (Use-FFN)	0.408	0.411	0.413	0.425	0.423	0.480	0.397
De-B (Skip-FFN)	0.530	0.521	0.563	0.507	0.562	0.594	0.508
Hybrid (Use-FFN)	0.499	0.352	0.452	0.425	0.425	0.483	0.410
Hybrid (Skip-FFN)	**0.571**	0.533	**0.576**	0.523	**0.587**	0.408	**0.573**
mBERT	0.490	0.508	0.502	0.488	0.494	**0.542**	0.514

MultiProp-Chunk							
Ensemble Models	**En**	**Ar**	**Ge**	**It**	**Fr**	**Po**	**Ru**
En-B(Use-FFN)	0.441	0.409	0.422	0.425	0.432	0.480	0.409
En-B (Skip-FFN)	**0.590**	0.569	0.579	0.496	0.572	**0.625**	0.469
De-B (Use-FFN)	0.436	0.449	0.436	0.437	0.421	0.477	0.410
De-B (Skip-FFN)	0.546	0.589	0.576	**0.503**	0.558	0.611	0.441
Hybrid (Use-FFN)	0.499	0.446	0.452	0.425	0.425	0.505	0.408
Hybrid (Skip-FFN)	0.567	**0.598**	**0.584**	0.457	**0.584**	0.547	**0.595**
kinit-sk	0.574	0.556	0.514	0.513	0.553	0.478	0.562

MultiProp-ML							
Ensemble Models	**En**	**Ar**	**Ge**	**It**	**Fr**	**Po**	**Ru**
En-B(Use-FFN)	0.454	0.438	0.441	0.425	0.433	0.483	0.328
En-B(Skip-FFN)	0.562	0.570	0.579	**0.579**	0.573	0.590	0.526
De-B(Use-FFN)	0.442	0.462	0.403	0.423	0.440	0.478	0.400
De-B(Skip-FFN)	0.512	**0.571**	0.569	0.500	0.554	**0.602**	0.491
Hybrid (Use-FFN)	0.499	0.395	0.425	0.425	0.425	0.480	398
Hybrid (Skip-FFN)	**0.573**	0.538	**0.583**	0.514	**0.587**	0.422	**0.583**
XLM-R	0.483	0.511	0.509	0.516	0.506	0.575	0.482

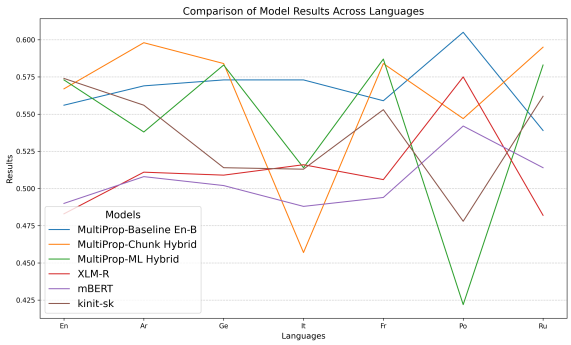

Table 2: Model Results Across Languages

Table 3: F1 Micro Scores of Our Three Proposed Systems across Seven Languages under a Few-Shot Learning Setting. The tables present the performance results of our models on datasets in English (En), Arabic (Ar), German (Ge), Italian (It), French (Fr), Polish (Po), and Russian (Ru). We implemented three different ensemble models: Encoder-Based (En-B), Decoder-Based (De-B), and Hybrid. Each model was tested using two methods for prediction aggregation: Use-FFN and Skip-FFN. The best score for each language is boldfaced.

nature, which aligns with label-specific linguistic patterns. Hybrid models, on the other hand, excel at identifying labels like *Doubt* and *Slogans*, leveraging the strengths of both encoder and decoder paradigms to handle mixed structural and semantic cues.

Benchmark models like XLM-R and mBERT exhibit stable performance but underperform compared to MultiProp models. While these state-of-the-art models provide consistent results, they lack the tailored architecture and cross-lingual adaptability inherent in MultiProp.

Figure 2 compares a selected sample of our models with state-of-the-art systems, including kinit-sk (Hromadka et al., 2023), which excelled in SemEval 2023 Propaganda Detection across various languages, as well as XLM-R Large and mBERT. Skip-FFN achieves superior F1-micro scores, excelling in low-resource settings, while Use-FFN performs better in F1-macro scores for rare labels. The MultiProp-Chunk Hybrid model surpasses kinit-sk in Arabic and Russian while remaining competitive in other languages. The MultiProp-Baseline En-B model excels in Polish and Italian, while the MultiProp-ML Hybrid model demonstrates consistent cross-lingual performance in English, German, French, and Russian. These results underline the advantages of tailored architectures for multilingual tasks.

7 Conclusion

In this work, we developed a robust multilingual framework by leveraging a range of pretrained models, ensembling techniques, and machine learning methods. Our approach combines multiple models to create a language-agnostic system that effectively understands and transfers knowledge across languages, with the addition of a monolingual model enhancing performance for the target language. By integrating multilingual embeddings with word embeddings and deploying a diverse set of classifiers, we achieved notable improvements across various languages. Specifically, our ensemble of advanced classifiers outperformed traditional stacking methods, resulting in a 13% increase in prediction accuracy. In future work, we aim to expand our dataset to include Abjad and Ajami languages, such as Persian and Pashto, and evaluate the scalability of our ensemble by incorporating language-specific monolingual models or relying solely on multilingual models.

8 Limitations of the work

Despite incorporating multiple languages such as Arabic, German, English, Italian, French, Polish,

and Russian, our dataset faces constraints due to the limited availability of annotated data for less-resourced languages, particularly Arabic. This limitation may affect the generalizability of the models to other low-resource languages not included in the dataset. Data augmentation techniques, including translation, were employed to enhance the dataset. However, the translation process might lead to the loss of nuanced labels related to specific propaganda techniques. The subtleties necessary for accurately detecting these techniques may not fully translate, potentially diminishing the effectiveness of the model. Additionally, the dataset exhibits class imbalance issues. For instance, the "Loaded Language" technique is frequently represented across many languages, while other techniques, such as "Presenting Irrelevant Data (Red Herring)" may have few or no samples in some languages like Russian. This imbalance complicates performance evaluation and is further impacted by the use of the F1 micro metric, which tends to favor majority classes and can obscure the model's performance on less-represented techniques.

Acknowledgments

This work has been supported by the German Federal Ministry of Education and Research (BMBF) as part of the DeFaktS program. Any opinions, findings, conclusions, or recommendations in this material are those of the authors and do not necessarily reflect the views of the BMBF. This research was also supported by the state of North Rhine-Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence. We would like to thank all the anonymous reviewers for their valuable input.

References

Sergiu Amihaesei, Laura Cornei, and George Stoica. 2023. Appeal for attention at semeval-2023 task 3: Data augmentation extension strategies for detection of online news persuasion techniques. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 616–623.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Md Rabiul Awal, Roy Ka-Wei Lee, Eshaan Tanwar, Tanmay Garg, and Tanmoy Chakraborty. 2023. Model-agnostic meta-learning for multilingual hate speech detection. *IEEE Transactions on Computational Social Systems*, 11(1):1086–1095.

Alberto Barrón-Cedeno, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Leo Breiman. 1996. Bagging predictors. *Machine learning*, 24:123–140.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

Tanmay Chavan and Aditya Kane. 2022. Large language models for multi-label propaganda detection. *arXiv preprint arXiv:2210.08209*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

David R Cox. 1959. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 21(1):238–238.

Giovanni Da San Martino, Alberto Barron-Cedeno, and Preslav Nakov. 2019. Findings of the nlp4if-2019 shared task on fine-grained propaganda detection. In *Proceedings of the second workshop on natural language processing for internet freedom: censorship, disinformation, and propaganda*, pages 162–170.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Moussa Kamal Eddine, Nadi Tomeh, Nizar Habash, Joseph Le Roux, and Michalis Vazirgiannis. 2022. Arabart: a pretrained arabic sequence-to-sequence model for abstractive summarization. *arXiv preprint arXiv:2203.10945*.

Neele Falk, Annerose Eichel, and Prisca Piccirilli. 2023. Nap at semeval-2023 task 3: Is less really more?(back-) translation as data augmentation strategies for detecting persuasion techniques. *arXiv preprint arXiv:2304.14179*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Yoav Freund, Robert E Schapire, et al. 1996. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer.

Ivan Habernal, Raffael Hannemann, Christian Pollak, Christopher Klamm, Patrick Pauli, and Iryna Gurevych. 2017. Argotario: Computational argumentation meets serious games. *arXiv preprint arXiv:1707.06002*.

Philipp Heinisch, Moritz Plenz, Anette Frank, and Philipp Cimiano. 2023. Accept at semeval-2023 task 3: An ensemble-based approach to multilingual framing detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1347–1358.

S Hochreiter. 1997. Long short-term memory. *Neural Computation MIT-Press*.

Timo Hromadka, Timotej Smolen, Tomas Remis, Branislav Pecher, and Ivan Srba. 2023. Kinitveraai at semeval-2023 task 3: Simple yet powerful multilingual fine-tuning for persuasion techniques detection. *arXiv preprint arXiv:2304.11924*.

Ali Haghpanah Jahromi and Mohammad Taheri. 2017. A non-parametric mixture of gaussian naive bayes classifiers based on local independent features. In *2017 Artificial intelligence and signal processing conference (AISP)*, pages 209–212. IEEE.

Venkatachalam Kandasamy, Pavel Trojovskỳ, Fadi Al Machot, Kyandoghere Kyamakya, Nebojsa Bacanin, Sameh Askar, and Mohamed Abouhawwash. 2021. Sentimental analysis of covid-19 related messages in social networks by involving an n-gram stacked autoencoder integrated in an ensemble learning scheme. *Sensors*, 21(22):7582.

Moonsung Kim and Steven Bethard. 2020. Ttui at semeval-2020 task 11: Propaganda detection with transfer learning and ensembles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1829–1834.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. *arXiv preprint arXiv:2005.00633*.

G Martino, Alberto Barrón-Cedeno, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. Semeval-2020 task 11: Detection of propaganda techniques in news articles. *arXiv preprint arXiv:2009.02696*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Shubham Mittal and Preslav Nakov. 2022. Iitd at the wanlp 2022 shared task: Multilingual multi-granularity network for propaganda detection. *arXiv preprint arXiv:2210.17190*.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. *arXiv preprint arXiv:2003.02739*.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 838–844. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multilingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rafet Sifa, Maren Pielka, Rajkumar Ramamurthy, Anna Ladi, Lars Hillebrand, and Christian Bauckhage. 2019. Towards contradiction detection in german: a translation-driven approach. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2497–2505. IEEE.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Kai Ming Ting and Ian H. Witten. 1997. Stacked generalizations: When does it work? In *International Joint Conference on Artificial Intelligence*.

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

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

A Appendix

A.1 Dataset Information After Merging Different Sets and Overlapping

Table 4: Instances per Language After Merging Different Sets and Overlapping

Language	Train	Dev	Test
ar	517	68	68
de	204	115	101
en	488	143	101
fr	164	95	101
it	273	142	101
ru	141	0	48
po	124	0	45

This table presents the number of instances in the dataset after merging different sets and handling overlapping data. The Arabic dataset ("ar") combines the original WANLP data with additional oversampled instances from the PTC translated data. For the languages de, fr, it, and en, the translated test set from PTC English was used for evaluation, while the development sets of po and ru were used as test sets to assess the model's performance on original language data. This setup allows for a comprehensive evaluation of the model's ability to generalize across both translated and original datasets.

A.2 Label Distribution across the Training Set of All Languages

Table 5 provides a detailed comparison of the distribution of various propaganda techniques (or labels) across different languages, including English, Arabic, German, Italian, and French, indicating the frequency of this propaganda technique in those languages.

A.3 Text Length Across Labels and Languages

Figure2 delves into the mean average text length for each label across the five datasets. Notably, Arabic texts exhibit significantly shorter lengths compared to their German, English, Italian, and French counterparts. This can be attributed to the Arabic dataset's composition, which includes a mix of articles and tweets, the latter being considerably shorter in length. Despite this, the overall trend shows that labels such as "Straw Man," "Thought-Terminating Cliché," and "Causal Oversimplification" consistently feature longer text lengths across all languages. Moreover, German articles stand out for having the most extended text lengths when compared to other languages, reflecting the nature of the content. An important observation is that text lengths across all datasets exceed the 512-token limit, which is the maximum sequence length that many models can process effectively. Specifically, the text lengths in our datasets range from 2,000 to 12,000 tokens. This significant discrepancy was a key motivation behind the development of the MultiProp-Chunk model, designed to handle longer sequences by breaking them into manageable chunks, ensuring that the entirety of the text can be processed without losing critical information.

A.4 Topic Modeling and Thematic Classification Across Multilingual Datasets

To analyze the topics in our dataset, we first preprocessed the text data by tokenizing, lemmatizing, and removing stopwords to standardize the input. We then applied Latent Dirichlet Allocation (LDA) [5], specifically using the LdaMulticore model from Gensim, to extract seven distinct topics from each language dataset. Using the Bag of Words representation, we identified key terms associated with these topics. Subsequently, we categorized these topics into overarching thematic groups based on their content, resulting in a clear classification of themes such as "Political Discussions, Elections" and "COVID-19". This approach enabled a comprehensive understanding of the primary themes present across different languages in the dataset.To visualize the topic distribution across the datasets, we generated pie charts for each language 3

B Technical Details

B.1 Weighted Loss Function for Multi-Label Classification

The weighted loss function takes the form:

[5] https://github.com/piskvorky/gensim

Label Distribution after applying Data Augmentation Techniques					
Labels	**English**	**Arabic**	**German**	**Italian**	**French**
Presenting Irrelevant Data (Red Herring)	52	16	34	34	43
Loaded Language	413	404	147	255	157
Thought-terminating cliché	119	42	89	121	85
Exaggeration/Minimisation	249	118	113	129	110
Repetition	199	64	45	51	48
Slogans	129	65	71	55	73
Flag-waving	177	48	65	47	30
Doubt	238	100	144	224	118
Appeal to authority	122	46	88	67	52
Bandwagon	45	30	39	34	51
Causal Oversimplification	133	62	61	67	68
Obfuscation, Intentional vagueness, Confusion	44	18	37	27	63
Name calling/Labeling	323	269	179	205	131
Reductio ad hitlerum	68	83	59	47	63
Appeal to fear/prejudice	204	132	101	148	90
Whataboutism	33	37	36	39	52
Black-and-white Fallacy/Dictatorship	102	51	57	62	55
Misrepresentation of Someone's Position (Straw Man)	34	24	23	35	65

Table 5: Label Counts Across Different Languages

$$\mathcal{L} = -\sum_{i=1}^{C} w_i \left[y_i \cdot \log(\sigma(z_i)) + (1 - y_i) \cdot \log(1 - \sigma(z_i)) \right]$$

where C is the number of classes, w_i is the weight for class i, y_i is the true label, z_i is the raw output (logit) from the classifier, and $\sigma(\cdot)$ is the sigmoid function. This weighted loss function helps the model correctly predict multiple labels for a given text, especially for the less frequent classes.

Category	**Model**	**Learning Rate**
E-B	ARBERT	1e-5
	bert-base multilingual-cased	3.7e-6
	bert-base-cased	5e-5
	xlm-roberta-large	4.4e-6
D-B	aragpt2-base	2e-5
	gpt2-large	1.8e-5
	gpt2-medium	1.8e-6
Hybrid	AraBART	3e-5
	mt5-large	2e-5
	mbart-large-50	3e-6

Table 6: Models and Their Learning Rates

C Hyper-parameters

C.1 Ensemble Models Parameters

Through extensive experimentation, we determined the optimal learning rates for each model, in line with established best practices and recommendations for BERT, XLM-R, and GPT-2 models.

Regarding training specifics, we found a batch size of 10 to be optimal, with loss improvement plateauing after 4 epochs. We set the tokenization length to 512 to balance context capture with memory constraints and applied a dropout rate of 0.1 to mitigate overfitting. The AdamW optimizer was used across all ensemble models due to its efficacy with transformer architectures and handling sparse gradients. Additional key hyperparameters are detailed in Table 7.

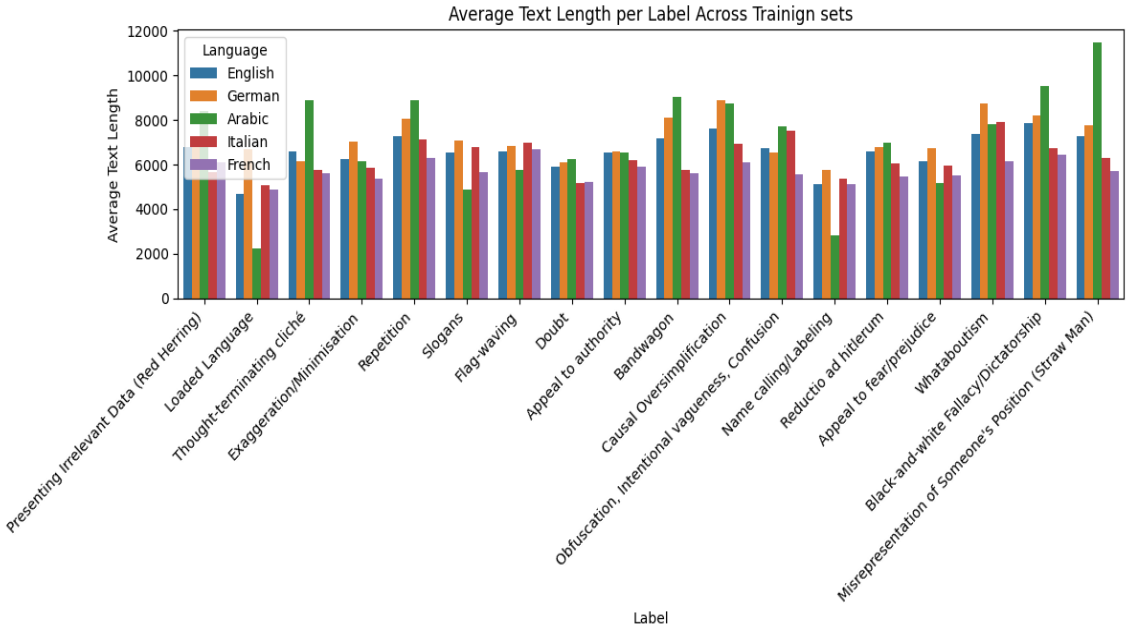

Figure 2: Average Text Length per Label

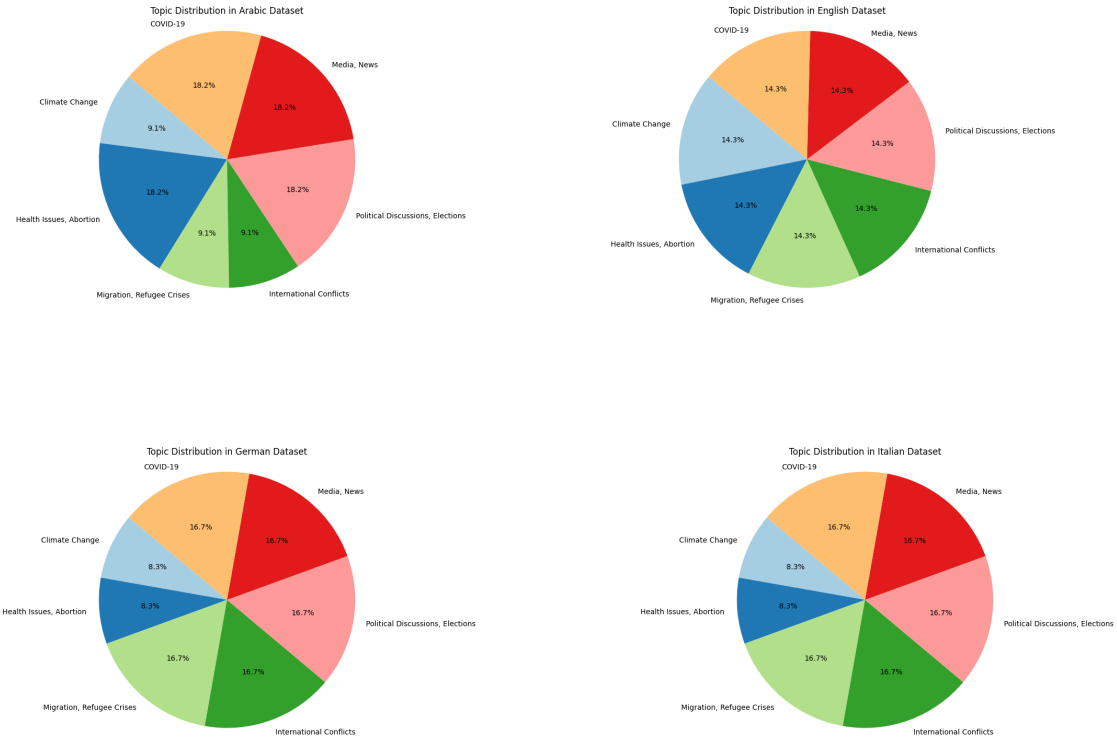

Figure 3: Pie charts illustrating the distribution of topics across various languages.

C.2 Meta Estimators Parameters

For predictions aggregation, we utilized several advanced classifiers and meta-estimators. The parameters for these classifiers, optimized using grid search with cross-validation (cv = 5) on the development sets, are summarized in Table 8.

Parameter	Value
Meta Models Learning Rate	2e-5
Maximum Gradient Norm	1.0
Number of Labels	18
Embedding Dimension	1024
Max sequence length	512
Overlap	100
Threshold for FFN Prediction	0.5
Threshold for Classifiers Prediction	0.35
Learning Approach	'zero_shot' or 'few_shot'
Maximum Chunks	25

Table 7: Hyperparameters and Additional Parameters

Classifier	Parameters

Random Forest	`n_estimators=100`
	`criterion='gini'`
	`bootstrap=True`
	`oob_score=True`
	`random_state=0`
	`max_features='sqrt'`
	`class_weight='balanced'`

Gaussian NB	Used with `ClassifierChain` due to multilabel classification `var_smoothing=1e-07`

Logistic Regression	Used with `ClassifierChain` due to multilabel classification
	`solver='liblinear'`
	`C=0.1`
	`class_weight='balanced'`
	`penalty': 'l1'`

SVM	Used with `OneVsRestClassifier` for multilabel classification
	`kernel='poly'`
	`C=1.0`
	`decision_function_shape='ovr'`
	`class_weight='balanced'`

xgboost	`n_estimators=100`
	`learning_rate=0.1`
	`max_depth=3`
	`random_state=0`

Table 8: Models and their parameters used in our evaluation

D Additional Results

D.1 Performance Evaluation of Different Embedding Methods for Ensemble Models

In this study, we utilize the F1 score to evaluate the performance of three ensemble models: encoder-based, decoder-based, and hybrid. These models are assessed using three distinct embedding methods:

1. Transformer-based Embedding: We extract embeddings from transformer models and concatenate them within the ensemble.

2.Transformer-based + GloFast Embedding: Transformer-based embeddings are combined with GloFast embeddings, which integrate GloVe and FastText features.

3. Transformer-based + TF-IDF Embedding: We calculate TF-IDF across the dataset and concatenate it with transformer-based embeddings for each instance.

Our aim is to identify the most effective embedding method, which can then be used as the default for all ensemble models. The experiments, conducted in a few-shot setting, are presented in Table 9 , with bolded values representing the highest F1 micro scores for each language and embedding method. Additionally, GloFast shows significant potential for further improvement. By increasing the embedding dimensions from 200 (100 from GloVe and 100 from FastText) to 600 (300 for each model), we expect to enhance performance. We also believe that expanding the training dataset to include languages like Italian and French will further boost results. Although GloFast was initially trained only on Arabic, English, and German, its ability to generalize across languages and effectively handle out-of-vocabulary words using the "unknown" vector demonstrates its versatility.

While GloFast consistently performs well, the combination of TF-IDF with transformer-based models has delivered particularly strong results for Italian and French. In contrast, transformer-based embeddings alone achieved the highest scores for German when used with the encoder-based

ensemble model. This success can be attributed to the pretrained multilingual models and the specific nature of the German dataset, which combines SemEval23 data with oversampled instances from the translated PTC dataset.

D.2 Comparison of Approaches in Zero-Shot Setting

The results in Table 10 highlight the performance of our MultiProp Framework, which consists of three components: MultiProp-Baseline, MultiProp-Chunk, and MultiProp-ML, evaluated across seven languages: English, Arabic, German, Italian, French, Polish, and Russian. In a zero-shot setting, we trained and fine-tuned the models on English and German, then assessed their ability to generalize to the other languages. Similar to the few-shot experiment, each system was evaluated using three ensemble architectures: Encoder-Based (En-B) models like mBERT and XLM-R, Decoder-Based (De-B) models such as GPT-2 Large, and Hybrid models like mBART and mT5. For each component, we applied two prediction aggregation methods: Use-FFN and Skip-FFN.

Our baseline model demonstrated strong performance in languages like Italian, Polish, and Russian, even though no training data from these languages was used, validating the model's generalization capabilities in a zero-shot setting. Notably, Skip-FFN outperformed Use-FFN in most cases; however, in Polish, the Use-FFN method showed better performance with hybrid ensemble models (mBART and mT5) in both the MultiProp-Chunk and MultiProp-Baseline architectures, indicating its effectiveness with encoder-decoder-based models for Polish in the zero-shot setting.

The MultiProp-Chunk model further improved upon the baseline in many languages, including Polish, Russian, and French, when using the Skip-FFN method. Meanwhile, the MultiProp-ML model consistently outperformed the others in low-resource languages, showcasing its ability to transfer knowledge from high-resource languages in the zero-shot setting. It was especially effective in Arabic, where we leveraged a meta-learning approach with pseudo-labels to enhance performance.

Since all three models were trained on English and German, their performance on these languages remained consistent with the few-shot setting. As expected, our models outperformed state-of-the-art models like XLM-R and mT5 in the zero-shot setting across all languages, demonstrating the effectiveness of ensemble models and the integration of different embedding approaches and classifiers.

F1 Micro Scores of Our Systems and State-of-the-Art Models in Zero Shot							
MultiProp-Baseline							
Ensemble Models	**En**	**Ar**	**Ge**	**It**	**Fr**	**Po**	**Ru**
En-B (Use-FFN)	0.426	0.358	0.446	0.441	0.464	0.484	0.426
En-B (Skip-FFN)	0.562	**0.488**	0.574	**0.560**	0.524	**0.592**	**0.567**
De-B (Use-FFN)	0.431	0.369	0.427	0.449	0.445	0.515	0.445
De-B (Skip-FFN)	0.520	0.442	0.544	0.552	**0.541**	0.573	0.530
Hybrid (Use-FFN)	0.499	0.347	0.480	0.448	0.448	0.491	0.421
Hybrid (Skip-FFN)	**0.576**	0.377	**0.583**	0.494	0.502	0.447	0.446
MultiProp-Chunk							
Ensemble Models	**En**	**Ar**	**Ge**	**It**	**Fr**	**Po**	**Ru**
En-B(Use-FFN)	0.433	0.370	0.455	0.425	0.429	0.519	0.421
En-B (Skip-FFN)	**0.590**	**0.454**	0.573	0.511	0.503	**0.617**	**0.583**
De-B (Use-FFN)	0.435	0.335	0.422	0.429	0.424	0.476	0.426
De-B (Skip-FFN)	0.545	0.393	**0.576**	0.512	**0.566**	0.552	0.531
Hybrid (Use-FFN)	0.440	0.360	0.432	0.447	0.476	0.569	0.428
Hybrid (Skip-FFN)	0.568	0.232	0.560	**0.526**	0.434	0.379	0.547
MultiProp-ML							
Ensemble Models	**En**	**Ar**	**Ge**	**It**	**Fr**	**Po**	**Ru**
En-B(Use-FFN)	0.499	0.370	0.443	0.392	0.392	0.491	0.441
En-B(Skip-FFN)	0.567	**0.501**	**0.569**	**0.573**	**0.566**	**0.613**	**0.587**
De-B(Use-FFN)	0.430	0.359	0.412	0.427	0.431	0.479	0.431
De-B(Skip-FFN)	0.506	0.347	0.553	0.541	0.546	0.573	0.521
Hybrid (Use-FFN)	0.533	0.381	0.452	0.448	0.448	0.513	0.478
Hybrid (Skip-FFN)	**0.579**	0.432	**0.569**	0.538	0.474	0.535	0.502
State of the Art Models							
Baseline Models	**En**	**Ar**	**Ge**	**It**	**Fr**	**Po**	**Ru**
mBERT	0.351	0.263	0.347	0.336	0.353	0.388	0.337
mt5-large	0.334	0.295	0.358	0.341	0.341	0.380	**0.375**
gpt2-large	0.348	0.300	0.339	**0.365**	0.342	0.368	0.332
Llama2	0.341	0.278	**0.369**	0.341	0.331	**0.402**	0.337
XLM-R	**0.361**	**0.315**	0.324	0.338	**0.350**	0.375	0.343
mbart-large	0.351	0.310	0.336	0.348	0.354	0.371	0.350

Table 10: F1 Micro Scores of Our Three Proposed Systems across Seven Languages under a Zero-Shot Learning Setting.

Embedding Methods	F1 Micro Score of our three models using the Skip-FFN method							
	Models	**English**	**Arabic**	**German**	**Italian**	**French**	**Polish**	**Russian**
Transformer-based Embedding	**Encoder-Based**	0.546	0.543	**0.582**	0.582	0.566	0.590	0.509
	Decoder-Based	0.530	0.477	0.563	0.521	0.554	0.566	0.502
	Hybrid	0.570	0.531	0.577	0.547	**0.595**	0.431	0.552
Transformer-based+GloFast Embedding	**Encoder-Based**	0.556	**0.569**	0.573	0.573	0.559	**0.605**	0.539
	Decoder-Based	0.530	0.521	0.563	0.507	0.562	0.594	0.508
	Hybrid	**0.571**	0.533	0.576	0.523	0.587	0.408	**0.573**
Transformer-based+TF-IDF Embedding	**Encoder-Based**	0.560	0.551	0.575	**0.593**	0.560	0.590	0.562
	Decoder-Based	0.537	0.498	0.559	0.518	0.558	0.594	0.534
	Hybrid	0.570	0.530	0.578	0.509	**0.595**	0.433	0.541

Table 9: F1 Micro Scores for Different Embedding Methods and Ensemble Models