Zero and Few-Shot Learning Techniques for Cross-lingual Classification Tasks on Arabic and Code-Switched Data

Anonymous ACL submission

Abstract

Zero-shot and few-shot learning techniques offer promising solutions for addressing data scarcity in Natural Language Processing (NLP), particularly in under-resourced languages such as Arabic and code-switching scenarios. Traditional supervised deep learning methods often struggle in such contexts due to their dependence on extensive labeled data. In this paper, we propose a novel approach that utilizes zero-shot and few-shot learning methodologies for cross-lingual classification tasks, focusing on Named Entity Recognition (NER) in Arabic texts and sentiment analysis in both Arabic and code-switched Arabic-English data. We introduce two approaches, employing Pattern Exploiting Training (PET) and Better-few-shot learning in language models (LM-BFF), which demonstrate versatility across diverse classification tasks. Subsequently, we conduct comprehensive evaluations on NER and sentiment analysis tasks, showcasing the superior performance of LM-BFF, surpassing previous techniques by 1.5% f1-score in sentiment analysis of code-switched data. This study emphasizes the importance of zero and few-shot learning methodologies in overcoming data scarcity challenges in Arabic NLP and code-switching research, thereby advancing NLP capabilities in under-resourced linguistic contexts.

1 Introduction

Conventional supervised deep learning models in Natural Language Processing (NLP) traditionally rely on large annotated datasets for training, a requirement that becomes particularly challenging in under-resourced languages like Arabic and complex linguistic environments such as code-switching. Code-switching, the act of fluidly alternating between languages within a conversation, is a common phenomenon in multilingual communities. However, research on NLP for code-switching and Arabic lags behind that of well-resourced languages like English. This lack of data for code-switching and Arabic presents a significant hurdle for developing robust NLP models. However, addressing these challenges has led to the exploration of innovative learning paradigms such as zero-shot and few-shot learning (Xian et al., 2017). Zero-shot learning involves training a model to recognize classes that it has never seen during training, while few-shot learning focuses on learning from a limited number of examples per class (Wang et al., 2019). These approaches alleviate the need for extensive labeled data, making them particularly suitable for resource-constrained scenarios. Despite their efficacy, challenges persist in effectively addressing the complexities of code-switching and under-resourced languages (Balam, 2021). To bridge this gap, one widely used approach is transfer learning, where knowledge gained from one task or domain is utilized to improve performance on another task or domain (Brownlee, 2017). In scenarios with limited annotated data traditional transfer learning methods may not suffice. Herein lies the relevance of techniques like Knowledge Distillation and Auxiliary Language Model Training (Prottasha et al., 2022). Knowledge Distillation involves transferring knowledge from a large, well-trained model (teacher) to a smaller model (student), enabling the student model to generalize better in data-scarce environments (Hinton et al., 2015). Similarly, Auxiliary Language Model Training leverages data and annotations from related tasks or languages to enhance performance on the target task (Zhang et al., 2020). These methods reduce the burden of data annotation and extend the applicability of deep learning models. Our study investigates zero-shot and few-shot learning methodologies for Arabic NLP tasks, particularly focusing on Named Entity Recognition (NER) and Sentiment Analysis. NER involves identifying and categorizing entities such as names, locations, and organizations within text, while sentiment analysis aims to understand the expressed sentiment, providing valuable insights (Li et al., 2020; Tan et al., 2023). We explore zero-shot and few-shot learning techniques as flexible solutions for classification tasks in under-resourced languages like Arabic and code-switching contexts. Utilizing transfer learning, notably through approaches such as Pattern Exploiting Training (PET) (Schick and Schütze, 2020a,b) and Better-few-shot learning in language models (LM-BFF) (Gao et al., 2020), we elaborate on our methodology, adapting these techniques to our tasks, and assess their performance on both monolingual Arabic and code-switched Arabic-English data. Through our evaluation, we demonstrate significant performance improvements, particularly with LM-BFF, highlighting the potential of these approaches in addressing data limitations and advancing NLP in diverse linguistic environments.

2 Related Work

Advancements in zero-shot and few-shot learning within language processing have been notable, particularly with the emergence of large-scale language models like GPT-3. These models have been evaluated on a wide range of tasks, including machine translation, question answering, and text summarization in many languages, including English, Spanish, French, and many others. While these models excel in various tasks, their extensive size poses usability challenges and environmental concerns. GPT-3 comprises 175 billion parameters, prompting researchers to explore alternative approaches to achieve comparable performance without such extensive models. Some are developing models with reduced parameter counts to maintain high-performance levels, enhancing model accessibility and sustainability (Brown et al., 2020).

Addressing these limitations, alternative methods are actively explored. PET (Pattern Exploiting Training) addresses the limitations of using large language models (LLMs) by using cloze questions and verbalizers to create large training datasets without extensive manual labeling that allows the model to infer the label from the context (Schick and Schütze, 2020a,b). This bridges the gap between supervised and unsupervised learning. PET's effectiveness is shown on various tasks within SuperGLUE, demonstrating its versatility (Schick and Schütze, 2020a,b). iPET, an iterative variant of PET, improves on PET by continuously learning from its mistakes. It trains on a dataset that grows with each training cycle, focusing on areas where the model previously struggled (Schick and Schütze, 2020a,b).

Another noteworthy approach is LM-BFF (better few-shot fine-tuning) which efficiently fine-tunes LLMs with minimal data. Unlike traditional methods, it uses prompts and task demonstrations during fine-tuning, achieving good results on few-shot tasks (e.g., sentiment analysis, question answering) in various languages while requiring less computation (Gao et al., 2020). This makes LM-BFF particularly useful for situations with limited labeled data or for deploying LLMs on resource-constrained devices.

Another method that was introduced is BitFit. BitFit is a method for fine-tuning LLMs like GPT. It works by modifying only a small part of the model, specifically the bias terms, to achieve a specific task (Zaken et al., 2021). This makes BitFit more efficient and requires less memory than traditional fine-tuning methods. Even with this limited modification, BitFit can achieve accuracy comparable to traditional methods, especially when there is not a lot of data available for training (Zaken et al., 2021).

Another research paper focused on the Arabic zero-shot few-shot learning problem. The research introduces a self-training method for Arabic sequence labeling tasks that utilize unlabeled dialectal data to improve performance on Named Entity Recognition (NER) and Part-of-Speech (POS) tagging (Khalifa et al., 2021). This method achieves state-of-the-art accuracy on various Arabic datasets, demonstrating its effectiveness in handling limited labeled data and diverse dialects (Khalifa et al., 2021).

Another approach addresses the problem of zero-shot NLU for code-switching (mixing languages) using multilingual code-switching data augmentation (Krishnan et al., 2021). By randomly translating English text into various languages and using multilingual datasets, the research explores how code-switching improves performance in languages like Hindi and Turkish (Krishnan et al., 2021). This method, especially effective for languages distant from English, achieves higher intent accuracy and slot F1 scores (Krishnan et al., 2021).

3 Methodology

We explored the effectiveness of applying zero-shot and few-shot learning techniques to Arabic and code-switching data, a domain often under-represented in NLP research. To address this gap, we implemented and customized two existing techniques, Pattern Exploiting Training (PET) and Better Few-Shot Fine-tuning for Language Models (LM-BFF), to accommodate the unique linguistic complexities of Arabic and code-switching. Our objective was to overcome the challenges posed by limited labeled data and enhance their performance in this specific domain. Our approach involved fine-tuning PET using Pattern-Verbalizer Pairs (PVPs) optimized for Arabic and code-switching, while LM-BFF underwent adjustments to effectively handle the diverse linguistic structures inherent in these languages.

3.1 Pattern Exploiting Training (PET)

PET tackles the challenge of limited labeled data by offering two approaches for model creation: the base PET model and its iterative variant, iPET. The first approach that we used is PET. PET leverages human-provided knowledge through Pattern-Verbalizer Pairs (PVPs). PVPs consist of cloze-style questions that specifically target the task at hand. These questions are crafted using patterns designed to guide the model towards the relevant information within the data. For instance, a pattern for sentiment analysis might be "The movie was [MASK]. Overall, it was [MASK] experience." Here, the model would predict the missing sentiment words ("wonderful" and "positive") based on the context of the sentence.

PET goes beyond simple cloze questions by incorporating two key elements: ensembles of models and unlabeled data. First, PET utilizes an ensemble of Masked Language Models (MLMs). These individual models are trained on the cloze-question transformed data, allowing them to learn task-specific patterns. During a subsequent knowledge distillation step, the models learn from each other, collectively improving their performance. Second, PET leverages unlabeled data to further enhance its capabilities. This unlabeled data is transformed using the same patterns as the labeled data, providing additional context for the models during training. This process helps mitigate the risk of overfitting on the limited labeled data.

The second approach we employed is iterative PET (iPET) to address the challenge of zero-shot learning, where labeled data for some classes might be entirely absent. iPET tackles the scenario where even labeled data for some classes might be entirely absent. In iPET, we employ multiple generations of models. The first generation trains solely on patterns and unlabeled data, establishing a baseline performance. Subsequent generations leverage the previous generation's predictions on unlabeled data to create a new training dataset. These new training dataset are then used to progressively expand the original training dataset and refine the model's understanding of the task across generations. By utilizing PET's patterns, training strategy, and using iPET, we aimed to make PET more effective for the unique challenges posed by Arabic and code-switching data.

3.2 Better Few-Shot Fine-tuning for Language Models (LM-BFF)

The third approach that we used is LM-BFF which stands as a novel approach in NLP, particularly tailored to tasks necessitating effective adaptation with limited labeled data. The main idea of LM-BFF lies in its ability to leverage large pre-trained language models, such as BERT or GPT, and fine-tune them for specific downstream tasks. This methodology introduces three distinctive fine-tuning strategies, each catering to varying degrees of labeled data availability and task complexity. The first approach, conventional fine-tuning, follows the traditional paradigm of adapting the pre-trained model parameters to the target task using labeled data. For instance, in a sentiment analysis task, the model can fine-tune its parameters in order to recognize small sentiment cues in text snippets, thereby enhancing its predictive accuracy.

LM-BFF offers innovative solutions for tasks with scarce labeled data. It uses prompt tuning, a method that uses natural language prompts to guide predictions, enabling effective generalization to tasks with minimal labeled data. This approach is particularly useful in sentiment analysis tasks, where prompts like "The next sentence is? [MASK] The food was great" can guide the model to make accurate predictions. LM-BFF also introduces prompt tuning with demonstrations, which adds an additional layer of supervision during fine-tuning. Demonstrations showcase correct behavior, aiding the model in making more informed predictions. This enhances its ability to generalize effectively, even in scenarios with sparse labeled data. In sentiment analysis, for instance, "The next sentence is? [MASK] The food was great. The next sentence is? negative The film was awful" thus this helps the model to understand what labels it is expected to predict and in what context.

4 Evaluation & Results

To address Arabic Named Entity Recognition, we employed the ANERcorp dataset, which provides curated Arabic text for training purposes (Benajiba et al., 2007). For Arabic sentiment analysis, we utilized the ArSATwitter dataset, containing annotated tweets for sentiment classification (Saad, 2019). Additionally, for English-Arabic code-switching sentiment analysis, we relied on the ArEnSA dataset, an in-house dataset offering a diverse range of mixed Arabic-English text from platforms such as Twitter and YouTube.

4.1 PET Method Evaluation

To evaluate the first approach (PET), we optimized hyperparameters such as pre-trained models, learning rates, gradient accumulation steps, reproducible seeds, and mappings, testing alternative values for enhanced performance. Initially, default hyperparameters were used, including a learning rate of 1.00E-05, gradient accumulation steps of 1, and a seed of 13. Subsequent tests explored alternative values. Additionally, the training datasets consisted of only 10 rows to evaluate our few-shot results.

4.1.1 Pre-trained Model

We started with evaluating different pre-trained models (roberta-base, albert-base-v2, arabert-base, arabert-twitter-base) which showed that arabert-twitter-base excelled, especially on Arabic tasks as shown in Table 1. This is likely due to its training on 60 million Arabic tweets, leading to a 10% improvement in understanding human-like sentences compared to arabert-base (Antoun et al.). It even performed well on the ArEnSA dataset, demonstrating strong multilingual capabilities.

Table 1: Comparison of Different Models on Each Dataset (PET)

Model	ANERCorp		ArSATwitter		ArEnSA	
	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc
roberta-base	0.158	0.192	0.539	0.540	0.333	0.356
albert-base-v2	0.098	0.223	0.438	0.521	0.412	0.488
arabert-base	0.210	0.281	0.528	0.544	0.365	0.469
arabert-twitter-base	**0.294**	**0.369**	**0.728**	**0.729**	**0.504**	**0.534**

4.1.2 Hyperparameters

After fine-tuning a pre-trained model, we optimized multiple hyperparameters (learning rate, gradient accumulation steps, random seed) on our three datasets to find the best combination that maximizes performance. This achieved an accuracy of 0.371 and an F1-score of 0.359 for ANERCorp (learning rate: 2.00E-05, gradient accumulation steps: 1, random seed: 21), an F1-score of 0.735 and accuracy of 0.735 for ArSATwitter (learning rate: 1.00E-05, gradient accumulation steps: 2, random seed: 13), and an accuracy of 0.631 and an F1-score of 0.584 for ArEnSA (learning rate: 5.00E-05, gradient accumulation steps: 1, random seed: 13).

4.1.3 Verbalizer

PET's performance relies on the verbalizer, which connects target labels to the model's vocabulary. The verbalizer should accurately capture the semantic meaning of labels while the model understands them. Using a larger verbalizer can improve performance by mapping multiple labels to a single category, capturing synonyms and linguistic variations. Significant improvements were observed across all datasets, with ANERCorp's F1-score and accuracy increasing significantly. ArEnSA saw the most significant boost, reaching an F1-score of 0.610 and an accuracy of 0.638. Even ArSATwitter, which already performed well, benefited, achieving an F1-score of 0.748 and an accuracy of 0.749 as shown in Table 2. These results emphasize the importance of a rich verbalizer in PET for improved performance across various tasks.

Table 2: Comparison Small and Large Verbalizer on Each Dataset (PET)

Verbalizer	Dataset					
	ANERCorp		ArSATwitter		ArEnSA	
	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc
Small	0.359	0.371	0.735	0.735	0.584	0.631
Large	**0.390**	**0.445**	**0.748**	**0.749**	**0.610**	**0.638**

4.1.4 Patterns

We then focused on pattern exploration. Patterns act as instructions for the model, influencing how it interprets and predicts labels in specific language contexts. We began with a single pattern to establish a baseline, but the choice of patterns significantly impacts the model's ability to perform well. There are three main pattern categories: null, prompt, and punctuation patterns.

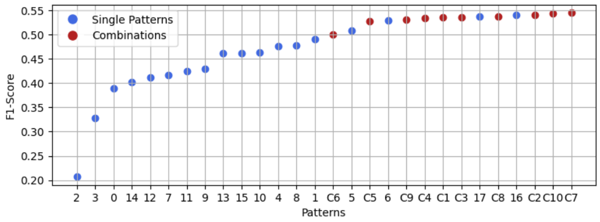

Figure 1: F1-Score for ANERCorp

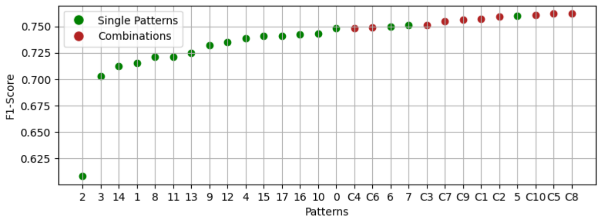

Figure 2: F1-Score for ArSATwitter

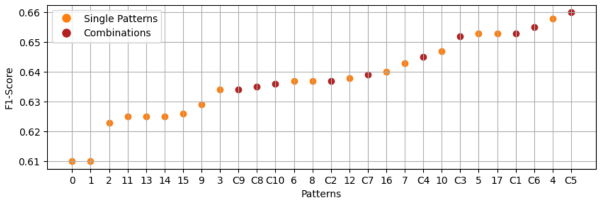

Figure 3: F1-Score for ArEnSA

Figure 4: Comparative F1-Score across Datasets Featuring 18 Distinct Patterns Ranging from 0-1 Null Patterns, 2-3 Prompt Patterns, and 4-17 Punc Patterns. The analysis extends to C1-C10 and explores the top 4 individual patterns in various combinations.

Table 3: Top 4 Patterns Results for each Dataset and Best Combination (PET)

Top 4 Patterns	Dataset					
	ANERCorp		ArSATwitter		ArEnSA	
	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc
P0	0.541	**0.575**	0.760	0.760	0.658	0.697
P1	0.538	0.561	0.751	0.752	0.653	**0.700**
P2	0.530	0.561	0.750	0.750	0.653	0.687
P3	0.509	0.528	0.748	0.749	0.647	0.681
Best Combination	**0.545**	0.568	**0.762**	**0.762**	**0.660**	0.698

While prompt patterns have shown success in other languages, we focused more on punctuation patterns due to challenges in designing effective prompts for complex Arabic sentences. We tested a total of 18 patterns (2 null, 2 prompt, and 14 punctuation) for each dataset. The top four performing patterns from each dataset are highlighted in Table 3. An example, for null patterns, is **"x [MASK]"** and for prompt patterns, is **"[MASK] الجملة السابقة؟ x"** and for punctuation patterns, is **"x? [MASK]"** where **x** represents the input sentence and **[MASK]** represents the label that the model will predict.

To further refine our approach, we went beyond individual patterns and explored combinations. We selected the top four patterns from all 18 tested (including null, prompt, and punctuation) and tested every possible combination. The best combination became the foundation for further testing. This meticulous selection ensures the chosen patterns effectively guide the model for superior performance.

For a visual representation of how pattern selection affects performance, see Figure 4. This figure shows scatter plots for F1-score across three datasets, encompassing the results of all 18 individual patterns and all combinations of the top four patterns. This visualization helps us understand the impact of both individual patterns and their combinations on the model's ability to adapt and perform well in various NLP tasks.

4.1.5 PET with Different Sizes

After determining optimal hyperparameters and pattern combinations, we explore how training dataset size affects PET models, crucial for understanding adaptability and scalability. Previous tests used a fixed size of 10 rows, but expanding to 10-100 rows shows how dataset size impacts PET's efficacy. This reveals PET's performance on larger datasets, insights into generalization, and capturing task nuances. Systematically increasing data size provides valuable insights into PET's robustness, revealing performance trends and potential limitations. Figures 5 and 3 visualize these trends.

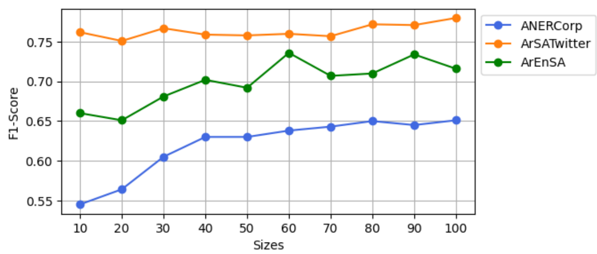

Figure 5: F1-Score for Different Sizes (PET)

4.2 iPET Method Evaluation

Secondly, we evaluate iPET. iPET, an extension of standard PET, employs an iterative training process with multiple model generations, each trained on datasets of increasing sizes. The methodology excels in distilling knowledge across generations, enabling subsequent models to benefit from collective insights. We will now explore how iPET performs in comparison with PET and see how it performs with different generation sizes, zero-shot, and different training dataset sizes

4.2.1 Different Generations

iPET builds on PET by training multiple generations of models on increasingly larger datasets. This iterative process lets each generation benefit from the knowledge of previous ones. We compared iPET's performance to PET's across different generation sizes, zero-shot learning tasks, and various training dataset sizes.

The evaluation involved four generation sizes with a fixed training dataset size. As expected, both F1-score and accuracy metrics consistently improved with more generations as shown in Table 4. This highlights the effectiveness of iPET's iterative refinement, where each generation builds upon the accumulated knowledge. This is particularly evident in the ANERCorp dataset, where metrics significantly improved across generations. This demonstrates the model's ability to learn and adapt through successive iterations.

Table 4: Comparison Between iPET Generations

iPET Generations	Dataset					
	ANERCorp		ArSATwitter		ArEnSA	
	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc
G1	0.526	0.544	0.758	0.758	0.655	0.698
G2	0.586	0.614	0.753	0.753	0.672	0.713
G3	0.596	0.631	0.753	0.753	**0.694**	**0.730**
G4	**0.602**	**0.636**	**0.767**	**0.768**	0.691	0.715

4.2.2 iPET Zero-shot

In zero-shot learning, iPET utilizes iterative knowledge accumulation to predict unseen classes without labeled examples. It draws on insights from previous generations to generalize to unfamiliar linguistic contexts. By employing the best pattern combination identified earlier, iPET demonstrates adaptability to evolving language, achieving positive results across all datasets: ANERCorp with an F1-score of 0.270 and accuracy of 0.307, ArSATwitter with an F1-score of 0.584 and accuracy of 0.655, and ArEnSA with an F1-score of 0.320 and accuracy of 0.405. These results underscore iPET's versatility and potential for handling unseen class challenges in NLP tasks.

4.3 LM-BFF Method Evaluation

Following the PET methodology, we adopted a parallel approach to optimize LM-BFF for our objectives, focusing on adjusting hyperparameters to match the unique characteristics of various datasets. This involved investigating key parameters such as learning rate, gradient accumulation steps, and seed values, starting with default settings of 1.00E-05, 1, 13, and 16 rows per label for training size. Initial tests assessed performance across datasets with these defaults, followed by further experiments to refine these values for enhanced flexibility and efficiency. This iterative process ensured LM-BFF's robustness and adaptability in different scenarios.

4.3.1 Hyperparameters

For the third approach, we concentrated on optimizing LM-BFF's hyperparameters, including learning rate, gradient accumulation steps, and random seed value, to achieve optimal performance for each dataset. The model was iteratively adjusted to improve flexibility and efficiency. The pre-trained model, arabert-twitter-base, was chosen for its effectiveness on Arabic datasets. Hyperparameter tuning yielded promising results, with F1-scores of 0.613 and 0.626 for ANERCorp (learning rate: 5.00E-05, gradient accumulation steps: 1, random seed: 13), 0.775 and 0.775 for ArSATwitter (learning rate: 5.00E-05, gradient accumulation steps: 1, random seed: 42), and 0.697 and 0.714 for ArEnSA (learning rate: 2.00E-05, gradient accumulation steps: 1, random seed: 13).

4.3.2 Types

Using LM-BFF, we tested three methods for model creation: prompts with demonstrations, prompts alone, and traditional fine-tuning, each addressing sequence classification tasks differently. Comparing results across datasets, "prompts with demonstrations" consistently outperformed others, with F1-scores listed in Table 5. Though "prompts alone" showed a slight improvement over fine-tuning, the difference was minimal. Traditional fine-tuning exhibited notably lower performance, emphasizing the effectiveness of incorporating prompts, especially those with demonstrations, for optimal sequence classification performance with LM-BFF.

Table 5: Comparison of Different Types on Each Dataset (LM-BFF)

Types	Dataset					
	ANERCorp		ArSATwitter		ArEnSA	
	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc
Prompt-demo	0.613	**0.626**	**0.779**	**0.775**	**0.697**	**0.714**
Prompt	**0.620**	0.622	0.773	0.767	0.690	0.688
Fine Tune	0.586	0.598	0.730	0.730	0.673	0.677

4.3.3 LM-BFF with Different Sizes

We conducted experiments on LM-BFF's performance with different dataset sizes, using three configurations: 8, 16, and 32 rows per label. Results showed consistent improvements in performance as training data size increased, indicating a clear trend in the model's handling as shown in Table 6.

Table 6: Comparison of Different Sizes for Each Dataset (LM-BFF)

Sizes	Dataset					
	ANERCorp		ArSATwitter		ArEnSA	
	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc
$num\_labels \times 8$	0.568	0.579	0.733	0.742	0.523	0.524
$num\_labels \times 16$	0.613	0.626	0.775	0.775	0.697	0.714
$num\_labels \times 32$	**0.670**	**0.686**	**0.819**	**0.819**	**0.730**	**0.744**

4.3.4 Patterns

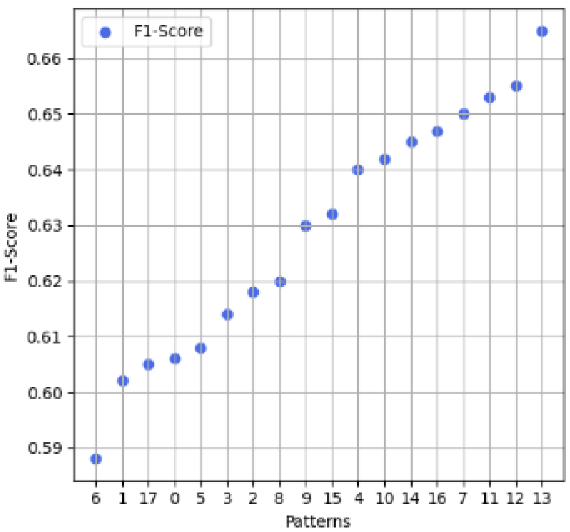

Figure 6: F1-Score for Different Templates

Table 7: Top 4 Templates Results for each Dataset (LM-BFF)

Top 4 Templates	Dataset					
	ANERCorp		ArSATwitter		ArEnSA	
	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc
T0	**0.665**	0.661	**0.783**	**0.783**	**0.783**	0.787
T1	0.655	0.672	0.780	0.780	0.779	**0.789**
T2	0.653	0.667	0.771	0.771	0.771	0.776
T3	0.650	**0.673**	0.769	0.770	0.755	0.763

LM-BFF uses templates like PET prompts to understand data, combining them with demonstrations. The same 18 templates, including null, prompt, and punctuation patterns, were used in PET experiments. Prompt patterns significantly improved performance compared to PET.

Figure 6 depicts the performance metrics associated with 18 templates, offering insights into how templates influence model behavior in sequence classification tasks. The top-performing templates for each dataset are outlined in Table 7, displaying their respective F1-scores and accuracies. These findings underscore the adaptability of LM-BFF and highlight the pivotal role of templates in refining its behavior for NLP tasks.

4.3.5 LM-BFF Zero-shot

LM-BFF demonstrates remarkable versatility in NLP tasks, particularly in zero-shot scenarios, where it encounters unseen classes using input patterns and verbalizers. Its prompt-based approach allows users to expand its capabilities without fine-tuning for each new class, relying on learned patterns for predictions. Table 8 illustrates the impact of top templates on zero-shot performance across datasets, underscoring the significance of template selection for optimal results.

Table 8: Zero-shot Results on different Templates and Traditional Fine Tuning (LM-BFF)

Zero-shot	Dataset					
	ANERCorp		ArSATwitter		ArEnSA	
	F1-Score	Acc	F1-Score	Acc	F1-Score	Acc
T0	0.243	0.327	**0.686**	**0.686**	0.172	0.312
T1	0.129	0.229	0.392	0.522	**0.296**	**0.358**
T2	0.235	0.307	0.594	0.606	0.226	0.334
T3	**0.287**	**0.384**	0.635	0.649	0.240	0.342

4.4 Final Results

Table 9: Comparison between different methods

Line	Examples	Methods	Dataset					
			ANERCorp		ArSATwitter		ArEnSA	
			F1-Score	Acc	F1-Score	Acc	F1-Score	Acc
1	T= 0	unsupervised	0.180	0.226	0.375	0.471	0.259	0.304
2		Fine-tuning	0.156	0.207	0.550	0.553	0.302	0.401
3		$LM - BFF_{prompt-demo}$	**0.287**	**0.384**	**0.686**	**0.686**	0.296	0.358
4		iPET	0.269	0.307	0.584	0.655	**0.320**	**0.406**
5	T= 10	supervised	0.166	0.187	0.470	0.501	0.480	0.566
6		Fine-tuning	0.295	0.314	0.691	0.691	0.599	0.604
7		$LM - BFF_{prompt-demo}$	0.410	0.453	0.693	0.706	**0.729**	**0.747**
8		PET	**0.545**	**0.568**	**0.762**	**0.762**	0.660	0.698
9	T= 100	supervised	0.149	0.221	0.624	0.625	0.677	0.697
10		Fine-tuning	0.638	0.646	0.819	0.829	0.815	0.820
11		$LM - BFF_{prompt-demo}$	0.650	0.666	**0.830**	**0.850**	**0.820**	**0.826**
12		PET	**0.651**	**0.707**	0.780	0.780	0.716	0.746
13	T= 500	supervised	0.459	0.495	0.757	0.757	0.833	0.839
14		Fine-tuning	0.689	0.696	0.881	0.881	0.855	0.859
15		$LM - BFF_{prompt-demo}$	0.683	0.693	**0.893**	**0.893**	**0.864**	**0.868**
16		PET	**0.707**	**0.727**	0.870	0.870	0.845	0.851
17	T= 1000	supervised	0.575	0.633	0.877	0.877	0.865	0.869
18		Fine-tuning	0.700	0.708	0.904	0.904	0.863	0.864
19		$LM - BFF_{prompt-demo}$	0.702	0.712	**0.914**	**0.914**	**0.875**	**0.878**
20		PET	**0.752**	**0.765**	0.905	0.905	0.873	0.878
21	Full Dataset	Previous SOTA	**0.860**	NA	NA	**0.970**	0.860	NA

In the final stages of evaluating the methods, we did an exhaustive investigation focused on identifying optimal templates and patterns for PET and LM-BFF techniques which can all be seen in Table 9. By meticulously selecting suitable templates and patterns for each dataset, the study achieved remarkable results, surpassing the previously established state-of-the-art (SOTA) results for the

ArEnSA dataset. The comparative analysis incorporated results from fine-tuning with Arabert-twitter, which consistently delivered optimal outcomes. Notably, the LM-BFF approach outperformed PET and traditional fine-tuning in zero-shot learning for ANERCorp and ArSATwitter datasets, achieving F1-scores of 0.287 and 0.686 for ANERCorp and ArSATwitter, respectively. Conversely, the ArEnSA PET method exhibited superior performance with an F1-score of 0.320 and an accuracy of 0.406.

In few-shot learning scenarios, PET demonstrated significant performance with limited data, achieving an F1-score of 0.545 and an accuracy of 0.568 for ANERCorp, and an F1-score and accuracy of 0.762 for ArSATwitter. The LM-BFF method proved optimal for ArEnSA, reaching an F1-score of 0.729 and an accuracy of 0.747. Upon expanding the training dataset to 100 and 500 rows, PET yielded peak performances for ANERCorp, while LM-BFF showed superior results for ArSATwitter and ArEnSA datasets. Remarkably, employing the LM-BFF method with a dataset size of 1000 instances yielded significant improvements, surpassing previous SOTA benchmarks for ArEnSA, achieving an F1-score of 0.875 and an accuracy of 0.878. Although falling slightly short for ANERCorp and ArSATwitter, the outcomes remained remarkably close to the SOTA benchmarks, showcasing the potential of the methods even with limited resources.

5 Conclusion and Future Work

In conclusion, our paper underscores the effectiveness of zero-shot and few-shot learning methods, notably PET and LM-BFF, in bolstering NLP models' adaptability to novel domains and tasks with minimal supervision. Through our exploration focused on Arabic language processing and code-switching challenges, we achieved significant advancements, surpassing previous benchmarks on the ArEnSA dataset.

Specifically, our experiments yielded notable results including an F1-score of 0.752 for the ANER dataset, an accuracy and F1-score of 0.914 for the ArSaTwitter dataset, and an impressive F1-score of 0.875 for the code-switched ArEnSA dataset, surpassing previous benchmarks by 1.5%.

Looking ahead, addressing computational constraints, refining linguistic techniques tailored for Arabic, exploring multilingual embeddings, and mitigating information loss from sentence truncation emerge as critical areas for future inquiry. By advancing research in these domains, we aim to propel Arabic NLP forward and cultivate robust natural language processing models capable of adeptly navigating diverse linguistic landscapes.

6 Limitations

The limitations of our study encompass several key challenges that influenced our approach and findings. Firstly, our reliance on Google Colab for conducting experiments posed significant constraints due to memory limitations and intermittent GPU availability. These factors resulted in delays and inefficiencies, particularly affecting the pace and reliability of our experimentation process. Despite these challenges, utilizing Colab was deemed necessary over local execution due to its practicality, albeit at the expense of optimal resource utilization.

Secondly, The complexity of Arabic and code-switched data presents significant challenges in developing effective patterns and verbalizers. Crafting precise mappings that balance specificity and generality is crucial for model robustness across diverse linguistic contexts. Existing pre-trained models have limitations in handling code-switching and limited labeled data, highlighting the need for improved multilingual embeddings and specialized pre-training techniques tailored to Arabic's linguistic characteristics.

Moreover, The model's ability to understand context was compromised by truncating input sentences, causing potential information loss. This compromise, while necessary for computational feasibility, could also reduce the accuracy of the models in predicting labels. These limitations suggest the need for future research to improve zero and few-shot learning techniques in cross-lingual classification tasks.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Osmer Balam. 2021. Beyond differences and similarities in codeswitching and translanguaging research. *Belgian Journal of Linguistics*, 35(1):76–103.

Yassine Benajiba, Paolo Rosso, and José Miguel Benedíruiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing: 8th International Conference, CICLing 2007, Mexico City, Mexico, February 18-24, 2007. Proceedings 8*, pages 143–153. Springer.

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.

Jason Brownlee. 2017. A gentle introduction to transfer learning for deep learning. *Machine Learning Mastery*, 20.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Muhammad Khalifa, Muhammad Abdul-Mageed, and Khaled Shaalan. 2021. Self-training pre-trained language models for zero-and few-shot multi-dialectal arabic sequence labeling. *arXiv preprint arXiv:2101.04758*.

Jitin Krishnan, Antonios Anastasopoulos, Hemant Purohit, and Huzefa Rangwala. 2021. Multilingual code-switching for zero-shot cross-lingual intent prediction and slot filling. *arXiv preprint arXiv:2103.07792*.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

Nusrat Jahan Prottasha, Abdullah As Sami, Md Kowsher, Saydul Akbar Murad, Anupam Kumar Bairagi, Mehedi Masud, and Mohammed Baz. 2022. Transfer learning for sentiment analysis using bert based supervised fine-tuning. *Sensors*, 22(11).

Motaz Saad. 2019. Arabic sentiment twitter corpus. Dataset on Kaggle.

Timo Schick and Hinrich Schütze. 2020a. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Timo Schick and Hinrich Schütze. 2020b. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Kian Long Tan, Chin Poo Lee, and Kian Ming Lim. 2023. A survey of sentiment analysis: Approaches, datasets, and future research. *Applied Sciences*, 13(7):4550.

Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. 2019. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–37.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4582–4591.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Linfeng Zhang, Muzhou Yu, Tong Chen, Zuoqiang Shi, Chenglong Bao, and Kaisheng Ma. 2020. Auxiliary training: Towards accurate and robust models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 372–381.