A Continual Pre-training Approach to Tele-Triaging Pregnant Women in Kenya

Wenbo Zhang,[1] **Hangzhi Guo,** [1] **Prerna Ranganathan,** [1] **Jay Patel,** [2] **Sathyanath Rajasekharan,** [2]
Nidhi Danayak, [1] **Manan Gupta,** [1] **Amulya Yadav** [1]

[1] Pennsylvania State University
[2] Jacaranda Health
wjz5120@psu.edu, hangz@psu.edu, pvr5208@psu.edu, jpatel@jacarandahealth.org, srajan@jacarandahealth.org,
nmd5606@psu.edu, mfg5480@psu.edu, amulya@psu.edu

Abstract

Access to high-quality maternal health care services is limited in Kenya, which resulted in ∼36,000 maternal and neonatal deaths in 2018. To tackle this challenge, Jacaranda Health (a non-profit organization working on maternal health in Kenya) developed PROMPTS, an SMS based tele-triage system for pregnant and puerperal women, which has more than 350,000 active users in Kenya. PROMPTS empowers pregnant women living far away from doctors and hospitals to send SMS messages to get quick answers (through human helpdesk agents) to questions about their medical symptoms and pregnancy status. Unfortunately, ∼1.1 million SMS messages are received by PROMPTS every month, which makes it challenging for helpdesk agents to ensure that these messages can be interpreted correctly and evaluated by their level of emergency to ensure timely responses and/or treatments for women in need. This paper reports on a collaborative effort with Jacaranda Health to develop a state-of-the-art natural language processing (NLP) framework, TRIM-AI (TRIage for Mothers using AI), which can automatically predict the emergency level (or severity of medical condition) of a pregnant mother based on the content of their SMS messages. TRIM-AI leverages recent advances in multi-lingual pre-training and continual pre-training to tackle code-mixed SMS messages (between English and Swahili), and achieves a weighted F_1 score of 0.774 on real-world datasets. TRIM-AI has been successfully deployed in the field since June 2022, and is being used by Jacaranda Health to prioritize the provision of services and care to pregnant women with the most critical medical conditions. Our preliminary A/B tests in the field show that TRIM-AI is ∼17% more accurate at predicting high-risk medical conditions from SMS messages sent by pregnant Kenyan mothers, which reduces the helpdesk's workload by ∼12%.

Introduction

Inadequate, low-quality health care is responsible for 50% of maternal deaths and 60% of neonatal deaths in hospitals around the world. This issue is particularly severe in Kenya, where the maternal mortality ratio (MMR) and neonatal mortality ratio (NMR) is estimated to be 362 and 3100 per 100,000 live births, respectively (as of 2020) (Scanlon et al. 2021; Maldonado et al. 2020). Unfortunately, more than 33% of maternal/neonatal deaths in Kenya occur due to delays in care-seeking, i.e., pregnant or puerperal women do not have (nearby) doctors available to take care of them at the right time. Thus, proper and timely decision making is a huge challenge for these women, especially as they lack access to timely information about their medical symptoms.

To address this issue, Jacaranda Health (a non-profit organization working towards improving maternity health in Kenya) developed PROMPTS, a widely used tele-triage tool that works as follows: (i) it enables pregnant or puerperal women to use a Short Message Service (SMS) based question-answer system to seek online health counseling and doctor assistance; (ii) once a pregnant woman registers for PROMPTS, she will start receiving "prompts" through text messages providing tips and behaviors based on her current stage of pregnancy. (iii) women can also use the question-answer function inside PROMPTS to ask questions regarding their health status by sending an SMS message to a toll-free phone number. (iv) A human helpdesk agent decides the case emergency level based on the content of the text message. (v) Finally, the helpdesk agent answers questions directly if the issue faced is simple and straightforward. On the other hand, urgent issues are promptly referred to the appropriate hospital or local clinics, so that mothers can get the necessary care at the right time.

PROMPTS has been successfully deployed in the field with ∼350,000 women, and has led to a marked improvement in the quality of health care for pregnant/puerperal women in Kenya (Health 2021). Unfortunately, it is very difficult to scale-up PROMPTS to larger populations. It currently receives ∼1.1 million SMS messages per month (28,000 new users enroll every month into PROMPTS), and the system currently relies on a small number of well-trained human helpdesk agents to evaluate case emergency levels and decide which cases should be answered directly and/or which cases need immediate medical intervention. Thus, these helpdesk agents are tasked with responding to thousands of incoming SMS messages every day. At this scale, it is very challenging for human agents to accurately identify the clinical urgency of the user's medical condition (based on their SMS message) to ensure that the most high-risk patients can be triaged first (to get timely care).

In this paper, we propose to automate the task of predicting emergency levels of a user's medical condition on the basis of their SMS messages. In particular, we formulate a multi-class classification problem which takes a user's SMS message as input, and generates as output a (softmax style) probability distribution over "intents" (each intent specifies a distinct category of medical symptoms/conditions that the user may be suffering from). This probability distribution is then used to calculate a weighted risk score for the input SMS message which specifies the emergency level of the user's medical condition. Unfortunately, solving this multi-class classification is challenging because of three reasons: (i) the input SMS messages received by PROMPTS are very noisy since they contain abbreviations, special symbols, slangs, emojis, etc. ; (ii) the input SMS messages are significantly code-mixed between English and Swahili, which makes it challenging to use BERT based NLP techniques (Dou et al. 2021); (iii) our NLP model should have very low inference times, as higher latencies would make PROMPTS unusable by pregnant women in need of urgent attention.

To tackle these challenges, we make three novel contributions. First, we develop a state-of-the-art natural language processing (NLP) framework for assessing emergency levels of SMS messages. Our NLP framework relies on a novel combination of recent advances in multi-lingual pre-training and continual pre-training to handle noisy code-mixed data. Our NLP model can potentially reduce the workload of helpdesk agents by providing them with a sorted batch of incoming SMS messages (in decreasing order of risk score), which enables them to triage patients in decreasing order of urgency. Second, we employ several techniques to simplify model structure and optimize the inference time of our prediction model (in order to meet Jacaranda Health's service level agreements). Third, our experimental evaluation shows that TRIM-AI significantly outperforms state-of-the-art baselines by achieving 0.774 weighted F_1 score on a ground-truth test set (for which the labels are annotated by well-trained helpdesk agents). Further, our inference time (for generating predictions for a single SMS message) on a single Nvidia Tesla K80 GPU is $\sim$28 milliseconds. These characteristics have enabled Jacaranda Health to improve their operational efficiency, while lowering the operational costs associated with maintaining PROMPTS.

Our framework has been reviewed by officials at Jacaranda Health and their feedback has been very positive. We have also worked with engineers and scientists from their team to integrate TRIM-AI into their existing codebase. Importantly, we have deployed TRIM-AI inside the PROMPTS platform as part of a real-world pilot study, which showed the effectiveness of TRIM-AI at accurately identifying high-risk conditions among pregnant women in Kenya. Based on the successful results from this pilot study, TRIM-AI has been in continuous deployment as part of the PROMPTS platform since June 2022.

Related Work

AI for Maternal Health. (Engelhard et al. 2018) uses an automated system to triage pregnant women in South Africa on the basis of SMS messages. However, they focus on examining violence against and mistreatment of women as the main high-priority label that needs to be predicted correctly. On the other hand, our work focuses broadly on the classification of incoming messages into several different intents (we have 58 different intents in our problem), and we leverage state-of-the-art advances in continual pre-training to develop our framework. In addition, Jacaranda Health has been using a preliminary AI triage bot to assess emergency levels of incoming SMS messages since 2018. However, their approach relies on a naive application of Google Translate API to translate all incoming code-mixed messages into mono-lingual English texts. This approach for handling code-mixing is highly limiting, as prior studies (Patil and Davies 2014) show that Google Translate works poorest ($\sim$10% accuracy) on Swahili-English translations (out of 26 different languages). Further, their AI bot is trained on an English-only SMS corpus by using off-the-shelf models from Google Cloud's Vertex AI platform. In our work, we address these limitations by handling code-mixed sentences via multi-lingual pre-training and continual pre-training, and leverage them to build a state-of-the-art NLP model for assessing emergency levels of SMS messages.

NLP Research for Code-Mixing with Low Resource Languages. To address challenges of working with non-English text data, XLM-ROBERTa (Conneau et al. 2019), a Transformer based masked language model trained on 100 different languages, was proposed. XLM-ROBERTa performs particularly well on low-resource languages, improving 15.7% in XNLI accuracy for Swahili. However, it is not explicitly designed to handle English-Swahili code-mixed data since the model is trained on independent monolingual data. Finally, (Gururangan et al. 2020) investigates the notion of continual pre-training in which a pre-trained model can be tailored to the domain of a target task (by using augmented data). They showed that a second phase of domain-adaptive pre-training leads to significant performance gains, with both high-resource and low-resource languages. In our work, we build upon these ideas by combining a multilingual pre-trained model (such as XLM-ROBERTa) with a continual pre-training framework (this part is pre-trained specifically with unlabeled SMS messages coming from PROMPTS) to develop a state-of-the-art NLP framework for assessing emergency levels of SMS messages.

Dataset

Code-mixed SMS messages within PROMPTS are stored across two different repositories. Before July 2021, PROMPTS used FreshDesk as its backend data management solution. Starting from July 2021, PROMPTS was migrated to SalesForce for their data management needs. We utilize both repositories of data.

FreshDesk Repository: This human-annotated repository contains 107,714 SMS messages (data points) received between October 2018 to July 2021. Each of these messages have been annotated with ground-truth intents (i.e., the target label) by well-trained helpdesk agents. There are 58 different types of intents inside this repository, each of which describes a category of medical symptoms (e.g., "*bleeding_abdomen*", "*pain_back*", etc.) that allows the PROMPTS system to determine the urgency of a patient's medical condition. In particular, each data point inside this repository

Example 1:

Code-mixed message: My son alimeza a coin since sunday na ajaenda haja is it risk ama itatoka tu

Translated English message: My son has swallowed a coin since sunday and that is going to need is it risky or it will just go away

Example 2:

Code-mixed message: Je heartburn inasababishwa na nini? sababu hata nikule nini lazma nikuwe nayo

Translated English message: What causes heartburn? Because even if I eat what I have to have

Figure 1: Modified examples of code-mixed sentences in the dataset (Blue and red text denotes English and Swahili phrases, respectively).

consists of the original SMS message (either in Swahili or in code-mixed format between Swahili and English) along with a ground-truth label (intent). Figure 1 shows two sample code-mixed data points from the FreshDesk repository.

For these intents, we apply a label encoder to map these intents into unique numeric keys. As a result, each data point in this dataset is associated with a number as its true intent label. We use this FreshDesk repository as our primary supervised dataset for training our NLP framework (as this is the only data with ground-truth labels). We use a stratified 80:20 split to divide this dataset into a training and test set.

SalesForce Repository: This repository contains 939,819 unlabeled SMS messages received since July 2021. Importantly, the ground-truth intent labels for these SMS messages is not known. Instead, the label for these SMS messages is derived using Jacaranda Health's preliminary AI triage bot (based on Google Vertex AI). As a result, we do not use these machine-generated labels for training our NLP framework (as otherwise, we would end up mimicking the Jacaranda Health's AI triage bot). Instead, we only use the SalesForce repository as an unlabeled dataset for continual pre-training inside our NLP framework.

TRIM-AI: Our NLP Framework

We now discuss the design of TRIM-AI in detail. We describe each component of our NLP framework separately (including pre-processing steps and model architecture).

Preprocessing of SMS Text Data

Noise in SMS messages is a known issue that has been investigated in a wide variety of previous work, with most of them focusing on language normalization (Dou et al. 2021). Unfortunately, little work has been done on normalization of code-mixed (Swahili and English) text. In this paper, we propose a general-purpose methodology that utilizes subword information for handling noise in such SMS messages.

The core step we apply for pre-processing code-mixed SMS text data is subword tokenization algorithms, as these algorithms are widely used to mitigate the out-of-vocabulary (OOV) problem in several downstream NLP tasks. In particular, we apply the Unigram segmentation algorithm (Kudo 2018) to perform subword segmentation directly on raw SMS data. Unigram segmentation is based on a probabilistic language model, and it assumes that each subword occurs independently, and consequently, the probability of a subword sequence is formulated as the product of the subword occurrence probabilities. The greatest benefit of this approach is that the probabilistic language model inside Unigram segmentation is language independent, and it also enables the modeling of noise explicitly. Thus, it can work well with low-resource languages like Swahili, and is also well suited to handle noise in code-mixed data.

In addition to Unigram segmentation, we also pad each sentence with zeros if they are shorter than the pre-defined maximum sentence length, and truncate sentences which are longer than the maximum sentence length. Note that we define the maximum sentence length as the threshold length value with which 95% of sentences in our dataset can maintain their original lengths.

Deep Learning Framework

TRIM-AI models the problem of assessing emergency risk levels of pregnant/puerperal women inside the PROMPTS platform (based on the contents of their SMS text messages) as a multi-class classification problem. The overall deep learning architecture and training procedure of TRIM-AI is shown in Figure 2. During training, TRIM-AI takes four inputs: (i) XLM-ROBERTa-large, a state-of-the-art multilingual pre-trained language model which is trained on a corpus of over 100 languages (including Swahili and English); (ii) the human-annotated FreshDesk repository containing all code-mixed SMS messages received before July 2021, along with their corresponding ground-truth intents; (iii) the unlabeled SalesForce repository containing all code-mixed SMS messages received after July 2021; and (iv) a predefined one-to-one mapping from intent to scalar risk scores, which is estimated by domain experts. Except for XLM-ROBERTa-large, the other three inputs used for training TRIM-AI are provided by Jacaranda Health.

At test time, TRIM-AI takes a single SMS message as input, and generates a probability distribution over all intents. With the help of our input intent-to-risk mapping, this probability distribution over intents is converted into a single scalar emergency risk score (signifying the severity of the user's medical condition), which is provided as output.

At a high level, the training procedure of TRIM-AI consists of two steps (Figure 2). First, we build XLM-ROBERTa-JH, a customized domain-specific multi-lingual pre-trained model by adapting XLM-ROBERTa-large to the domain-specific SalesForce repository of unlabeled SMS messages received by Jacaranda Health (or JH). Second, this domain-specific multi-lingual pre-trained model is then used as a building block inside the final TRIM-AI architecture. We discuss both these steps below.

XLM-ROBERTa-JH. In principle, off-the-shelf implementations of XLM-ROBERTa-large [1] should help in improving predictive performance on our multi-class classification

[1] An open-source pre-trained implementation of XLM-ROBERTa in Hugging Face

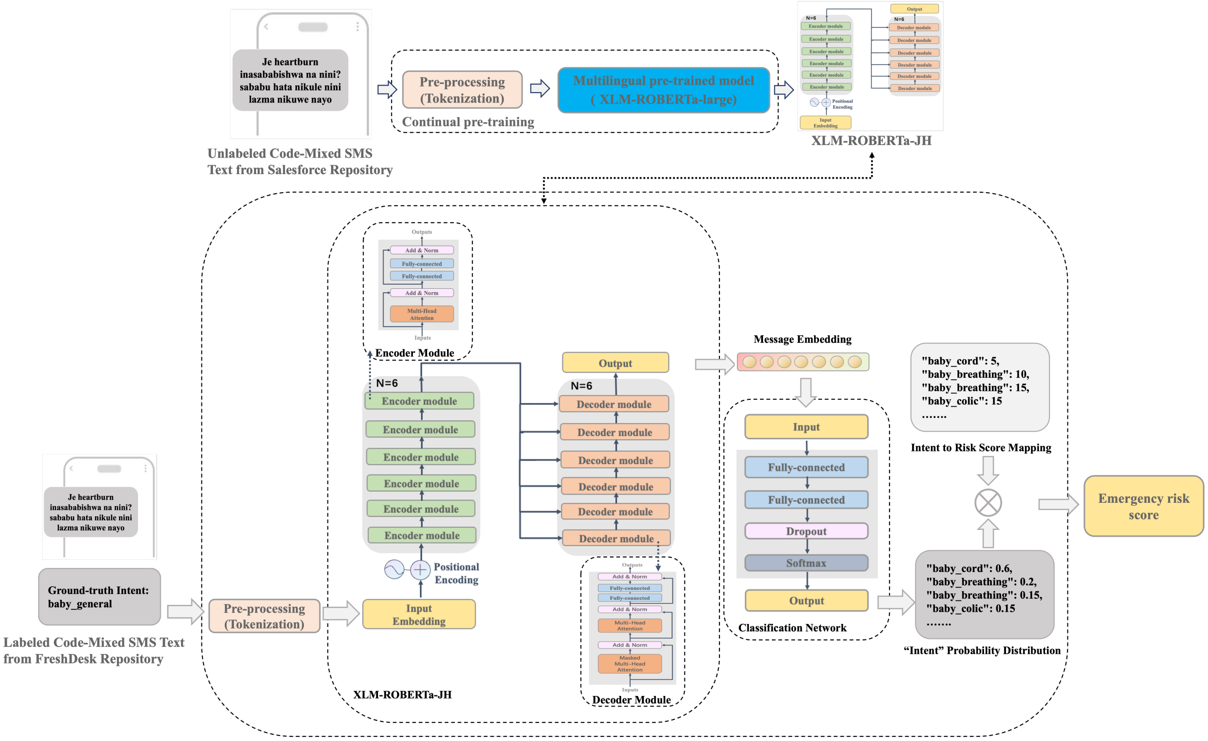
Je heartburn inasababishwa na nini? sababu hata nikule nini lazma nikuwe nayo
Unlabeled Code-Mixed SMS Text from Salesforce Repository
Pre-processing (Tokenization)
Multilingual pre-trained model (XLM-ROBERTa-large)
Continual pre-training
N=6
Encoder module
Decoder module
Output
Positional Encoding
Input Embedding
XLM-ROBERTa-JH
Outputs
Add & Norm
Fully-connected
Multi-Head Attention
Inputs
Encoder Module
Je heartburn inasababishwa na nini? sababu hata nikule nini lazma nikuwe nayo
Ground-truth Intent: baby_general
Labeled Code-Mixed SMS Text from FreshDesk Repository
Pre-processing (Tokenization)
Input Embedding
Masked Multi-Head Attention
Decoder Module
Message Embedding
Input
Fully-connected
Dropout
Softmax
Output
Classification Network
"baby_cord": 5, "baby_breathing": 10, "baby_breathing": 15, "baby_colic": 15 …….
Intent to Risk Score Mapping
"baby_cord": 0.6, "baby_breathing": 0.2, "baby_breathing": 0.15, "baby_colic": 0.15 …….
"Intent" Probability Distribution
Emergency risk score

Figure 2: Overall model architecture and training procedure of TRIM-AI.

task, as Swahili and English (the two languages present in our SMS text repositories) are present in the multi-lingual corpus of text used to train XLM-ROBERTa-large. Unfortunately, while XLM-ROBERTa works exceedingly well on sentences that are either completely in English or Swahili, its representation learning ability for code-mixed sentences containing both Swahili and English is still unsatisfactory (as confirmed by our experimental results).

Thus, to further improve XLM-ROBERTa-large's ability to deal with the particular type of code-mixed SMS messages contained with Jacaranda Health's repositories, we leverage continual pre-training techniques to fine-tune (or adapt) XLM-ROBERTa-large's off-the-shelf pre-trained model parameters to the target task (of predicting intents for code-mixed and noisy SMS messages). In particular, we attempt to enhance XLM-ROBERTa-large's cross-lingual ability by continually pre-training its language model with unlabeled code-mixed data from the SalesForce repository. Formally, (i) we begin by pre-processing SMS data from the SalesForce repository (via Unigram segmentation); (ii) the tokenized SalesForce data is then used to continually pre-train XLM-ROBERTa-large which adapts its parameters to the target domain (consisting of code-mixed SMS messages). (iii) the adapted model parameters after continual pre-training represents XLM-ROBERTa-JH, our domain-specific multi-lingual pre-trained model which is then used as a building block inside the final architecture of TRIM-AI.

Model Architecture. At a high level, TRIM-AI's model architecture is composed of three interconnected blocks. During training, labeled SMS messages from the FreshDesk repository are first passed through a pre-processing block, which tokenizes incoming SMS messages. Second, these tokenized messages are provided as input to XLM-ROBERTa-JH, our domain-specific pre-trained language model, which generates a dense vector embedding for the input SMS messages. This block enables us to further fine-tune the parameters of this pre-trained model via gradient descent back-propagation. Third, this dense vector embedding is passed as input into a classification network which generates as output a (softmax style) probability distribution over intents.

Classification Network. Our classification network is a feed-forward neural network consisting of two fully connected layers followed by a softmax layer to generate a probability distribution over intents for each input SMS message. The main role of the fully connected layers is to learn non-linear and high-dimensional vector embeddings which can help in generating accurate predictions. Finally, a single Dropout layer is also added to reduce the model generalization error during the training procedure.

Overall Algorithm. We now describe the steps of our overall algorithm. First, we apply pre-processing steps to the code-mixed SMS text contained in both the Freshdesk (labeled) and SalesForce (unlabeled) repositories. Second, we use the SalesForce dataset to apply continual pre-training on an off-the-shelf implementation of the XLM-ROBERTa-large model that results in an enhanced multilingual model which extends the model's representation ability for code-mixed text. This enhanced multilingual model is used as a building block in the final model architecture of TRIM-AI. Third, the FreshDesk labeled dataset is used to fine-tune the whole system consisting of the enhanced multilingual model and the classification network. We choose the Binary Cross-Entropy with Logit Loss (BCEwithLogitLoss) as our loss function to be optimized, and AdamW for model optimization (Loshchilov and Hutter 2017). Finally, we also use the linear warm-up mechanism (Gotmare et al. 2018) to ensure the stability of the training procedure during initial stages.

Experimental Evaluation

All experiments are run on a Debian based Deep Learning Image on the Google Cloud Platform with an Nvidia V100 GPU (unless specified otherwise). Our evaluation compares TRIM-AI against several state-of-the-art NLP baselines, and Jacaranda Health's preliminary Vertex AI model. We use the labeled *FreshDesk Repository* to evaluate the predictive performance of TRIM-AI and baseline models. We randomly split this *FreshDesk Repository* into an 80% train set and 20% hold-out test set. To compare the predictive performance of different models (TRIM-AI and baselines), we adopt weighted precision, recall, and F_1-scores as our evaluation metrics. These weighted metrics are well-suited for our evaluation setting because we deal with a multi-class classification problem with imbalanced labels. Finally, for each evaluated model, we average final results over three runs.

Baselines

We compare TRIM-AI against state-of-the-art baseline models, including four hierarchical neural network models and three pre-trained language models. In addition, we also compare with the performance of the Vertex AI model.

Hierarchical Neural Networks. To evaluate the effectiveness of TRIM-AI, we first compare our model with four systems that have a similar architecture. These systems all contain an input layer to preprocess original texts (into words, characters, or custom features), followed by an embedding layer that transforms processed texts into dense embedding vectors. These vectors are then further processed with an encoder layer to extract context and semantic information belonging to the original sentence. Finally, the output layer finishes the text classification task and generates corresponding labels. The only difference between these four systems is the structure of the encoder layer: (i) FastText which is designed for sentence-level text classification with n-gram features as model input (Joulin et al. 2016); (ii) TextRNN which utilizes RNN-based architectures to model text sequence with a multi-task learning framework (Liu, Qiu, and Huang 2016); (iii) TextCNN which applies convolutional neural networks built on top of word2vec to realize text classification (Kim 2014); (iv) RCNN which captures contextual information with recurrent structure and constructs the representation of text using a convolutional neural network (Lai et al. 2015).

Pre-trained Language Models (PLMs). In addition, we also compare against three ablations of TRIM-AI, each of which is created by replacing XLM-ROBERTa-JH inside TRIM-AI's model architecture (Figure 2) with alternate state-of-the-art pre-trained language models. *m-BERT* (Devlin et al. 2018) and *XLM-ROBERTa-base* (Conneau et al. 2019) are two state-of-the-art language models that are pre-trained on a corpus of multi-lingual data (which contains both Swahili and English text). Finally, Monolingual *RoBERTa* (Liu et al. 2019) is the backbone model of XLM-RoBERTa (which is our base architecture). Unlike *m-BERT* and *XLM-ROBERTa-base*, *RoBERTa*'s pre-trained data only contains the monolingual English corpus.

Vertex AI Model. We also compare TRIM-AI against the Google Vertex AI model which has been deployed by Jacaranda Health since 2018. The Vertex AI model uses Google Translate API to translate all incoming code-mixed messages into monolingual English text, and then uses an off-the-shelf black-box model from the Vertex AI platform to generate predictions.

Comparison Results & Analysis

Predictive Performance. Table 1 compares the weighted precision, recall and F_1 score of TRIM-AI and baseline models. Crucially, this table shows that TRIM-AI outperforms baseline models by a large margin. In particular, TRIM-AI achieves $\sim$5% higher weighted metrics than XLM-RoBERTa-base (its closest competitor). In addition, TRIM-AI significantly outperforms hierarchical neural network models, as it achieves $\sim$26%, $\sim$15%, $\sim$15%, $\sim$13% higher weighted metrics (on average) than TextCNN, FastText, RCNN, and TextRNN, respectively.

Moreover, Table 1 illustrates the superior performance of PLM based baselines in this classification problem. Notably, the worst performing PLM (i.e., m-BERT) outperforms the best performing hierarchical neural network model (i.e., TextRNN) by $\sim$6.6% in terms of weighted F_1 score. This result demonstrates that PLMs have stronger representation capabilities which enables them to better extract semantic and syntax information from code-mixed sentences as compared to hierarchical neural networks, where the entire model is trained from scratch using labeled datasets.

Finally, we also observe the superior effectiveness of multilingual PLMs (over monolingual models) in dealing with code-mixed sentences. Table 1 shows that despite using the same model architecture (which indicates the same representation power), XLM-RoBERTa-large achieves slightly better weighted metrics ($\sim$1% improvement on average) than the monolingual RoBERTa model. This result indicates the importance of adopting multi-lingual pre-trained models for handling code-mixed datasets.

Comparison with the Vertex AI Model. Table 1 shows that TRIM-AI significantly outperforms the Vertex AI model by achieving a 15.4% higher weighted F_1 score. This result is significant from a real-world perspective, as it illustrates that TRIM-AI significantly improves upon the current modus operandi of Jacaranda Health, which can potentially improve the efficiency of their day-to-day operations.

Further, we conduct a per-class comparison of the predictive performance of TRIM-AI and the Vertex AI model. Table 3 compares the weighted precision scores achieved by these two models on the top-5 and bottom-5 intents. Note that the top-5 (and bottom-5) intents consist of an ordered set of five intents with the best (and worst, respectively) predictive performance gap (as measured in terms of weighted precision) between TRIM-AI and the Vertex AI model. For example, TRIM-AI achieves significantly higher weighted precision (0.711) than the Vertex AI model (0.242) on the "*diarrhoea*" intent, whereas Vertex AI is able to outperform TRIM-AI on the "*anc-visit*" intent (0.704 vs 0.765 in terms of weighted precision). Importantly, the results in Table 3 reveal that TRIM-AI drastically outperforms the Vertex AI model on critical (high-risk) intents, e.g., TRIM-AI achieves $\sim$20% and $\sim$30% higher precision than Vertex AI on the *bleeding* and *baby_fever_discharge* intents, respectively. On the other hand, TRIM-AI performs poorer than the Vertex AI model on less critical intents (e.g., *facility_visit* and *anc_visit*), and even on these less critical intents, the predictive performance gap is significantly smaller in favor of the Vertex AI model, i.e., 7.34% lower precision score on average across the bottom-5 intents. This per-class comparison with the Vertex AI model highlights that the TRIM-AI is more suitable for deployment, as it ensures significantly improved predictive performance on SMS messages corresponding to highly critical intent categories.

Effectiveness of Continual Pre-training. Next, we highlight the importance of adopting continual pre-training to adapt the XLM-ROBERTa-large model inside TRIM-AI (which results in the domain specific XLM-ROBERTa-JH pre-trained model). Specifically, we compare the predictive performance of TRIM-AI against an ablated model which does not use continual pre-training to adapt the XLM-ROBERTa-large model. Further, we analyze two additional ablations of TRIM-AI, which are created by replacing XLM-ROBERTa-large with the XLM-ROBERTa-base pre-trained model (with and without continual pre-training) inside TRIM-AI's model architecture. Table 2 shows that continual pre-training guarantees a performance boost when migrating a multi-lingual PLM (trained for handling independent monolingual data) to handle code-mixed data, as adopting continual pre-training improves the predictive performance of TRIM-AI ablations with XLM-ROBERTa-base and XLM-ROBERTa-large pre-trained models by $\sim$4% and $\sim$1.3% in terms of weighted F_1 score. Importantly, Table 2 shows that the continual pre-training procedure leads to a greater boost in the predictive performance of an XLM-ROBERTa-base based TRIM-AI model (as compared to an XLM-ROBERTa-large based TRIM-AI model). This further underscores the effectiveness of continual pre-training as this procedure could potentially enable smaller-sized PLMs to achieve the same predictive performance as larger models.

Pilot Deployment

In collaboration with Jacaranda Health, we have conducted a pilot deployment of TRIM-AI, and compared its performance in the wild against Jacaranda Health's Vertex AI model. Specifically, we conduct several A/B tests between TRIM-AI and the Vertex AI model to quantify the real-world benefit accrued by the deployment of TRIM-AI in the field.

Our randomized A/B tests consist of the following steps: (i) Both TRIM-AI and the Vertex AI model are integrated into the Jacaranda Health's codebase, so that they can be deployed in parallel within the PROMPTS platform; (ii) when the A/B test begins, incoming SMS messages are randomly assigned to the TRIM-AI pipeline or to the Vertex AI pipeline (this second pipeline also requires the expensive usage of Google Translate API). (iii) Once SMS messages are assigned to an arm of the study, they will be passed through the respective NLP models (TRIM-AI or Vertex AI) to generate predicted intents as outputs (along with a confidence level of the intent detection). (iv) When the NLP model detects an intent with a confidence level greater than 75%, an automated response to the end-user is triggered. The response contains clinically vetted information relating to each intent and is always the same for each intent, differentiated only by whether the mother is pregnant or has delivered. For example, if a pregnant woman asks a question about headaches and the NLP model detects the "headache" intent at a confidence level of 75% or greater, she will immediately receive an automated response relating to headaches during pregnancy. If, however, a mother who has already delivered asks the same question, she will receive information on headaches post-pregnancy. (v) Immediately after the automated response is sent, the user (i.e., pregnant woman) will be asked whether the information provided answered her question, she has the option to respond "Yes" or "No". If she responds "Yes", her ticket in Salesforce will be updated, closed, and removed from the helpdesk team's workload. If, however, she responds "No", doesn't respond at all, or responds with anything other than Yes/No, her ticket in Salesforce will be updated and the helpdesk team alerted to follow up immediately with her question. The Yes/No responses are saved for later analysis. (vi) Finally, we analyze and compare the percentage of Yes/No responses received in the TRIM-AI and Vertex AI arms of the A/B test. Intuitively, a higher percentage of Yes responses received for automated responses sent in the TRIM-AI arm of the A/B test would represent validation (from end-users in the real world) of the effectiveness of TRIM-AI in generating high-quality predictions which satisfy the needs and concerns of pregnant/puerperal women in Kenya.

A/B Test Results: We create two separate study arms: in the first arm, SMS messages are prioritized and ranked using TRIM-AI, whereas in the second arm, SMS messages are ranked according to the Vertex AI model. Note that we compare against the Vertex AI model as a baseline because that is the current modus operandi of Jacaranda Health.

For a period of two weeks, incoming SMS messages are randomly assigned to either the TRIM-AI or the Vertex AI arm of the study. In total, 5,323 SMS messages are received within this time period, our of which 2753 messages are assigned to TRIM-AI and the remaining 2570 messages are assigned to Vertex AI.

Figure 3 measures the percentage of "*Yes*" replies received from mothers on SMS messages processed by TRIM-AI for 5 select intents (due to space limitations, the full results are in the appendix), and compares it against Vertex AI. All results shows TRIM-AI is highly effective in generating high-quality predictions based on incoming SMS messages, as pregnant women find automated query responses sent in the TRIM-AI arm of the study to be

Table 1: Comparison of predictive performance of TRIM-AI and other baseline models.

Model Name	Weighted Precision	Weighted Recall	Weighted F_1-score
Hierarchical NN (FastText as the encoder layer)	0.678	0.668	0.669
Hierarchical NN (TextRNN as the encoder layer)	0.682	0.673	0.67
Hierarchical NN (TextCNN as the encoder layer)	0.638	0.629	0.626
Hierarchical NN (RCNN as the encoder layer)	0.679	0.667	0.663
TRIM-AI (monolingual ROBERTa-base)	0.730	0.729	0.728
TRIM-AI (m-BERT)	0.727	0.726	0.725
TRIM-AI (XLM-ROBERTa-base)	0.736	0.735	0.735
Vertex AI model	0.765	0.598	0.671
TRIM-AI	**0.775**	**0.775**	**0.774**

Table 2: Evaluating the impact of continual pre-training on the TRIM-AI framework.

Model Name	Continual pre-training	Weighted Precision	Weighted Recall	Weighted F_1-score
TRIM-AI	No	0.763	0.763	0.762
TRIM-AI	Yes	0.775	0.775	0.774
TRIM-AI (XLM-ROBERTa-base)	No	0.748	0.747	0.746
TRIM-AI (XLM-ROBERTa-base)	Yes	0.771	0.770	0.770

Table 3: Comparison (Weighted Precision) between TRIM-AI and the VERTEX AI model: top five rows are intents where TRIM-AI performs best.

Intent type	**TRIM—AI**	**Vertex AI**
diarrhoea	0.711	0.242
numbness	0.809	0.421
baby_fever_discharge	0.781	0.492
baby_acne_pimples	0.837	0.577
bleeding	0.783	0.596
blood_group	0.466	0.625
facility_visit	0.390	0.464
baby_milestone_teething	0.856	0.928
ifas	0.853	0.886
anc_visit	0.675	0.704

more helpful and informative. From Figure 3, TRIM-AI outperforms the Vertex model by 40% on *baby_jaundice* (a highly critical intent). In addition, TRIM-AI achieves 18.31%, 10.99%, 8.43%, 7.87% higher than the Vertex model on *baby_milestone_general*, *edd*, *ultrasound*, and *family_planning*, respectively. Except for *ultrasound*, questions are assigned almost equally for other 4 intents.

TRIM-AI is also more cost-effective. According to estimates received from Jacaranda Health, the monthly cost for PROMPTS management has been reduced from \$819 (\$637 for the Vertex AI model + \$182 for Google translation API) to \$632 for TRIM-AI. This translates to monthly savings of ∼\$200 (or ∼24,000 Kenyan Shillings), which could then be used by Jacaranda Health to finance other critical operations. According to a quote from a senior NGO official, “the response automation built on top of the TRIM-AI has helped in reducing the PROMPTS helpdesk’s workload by ∼12%”.

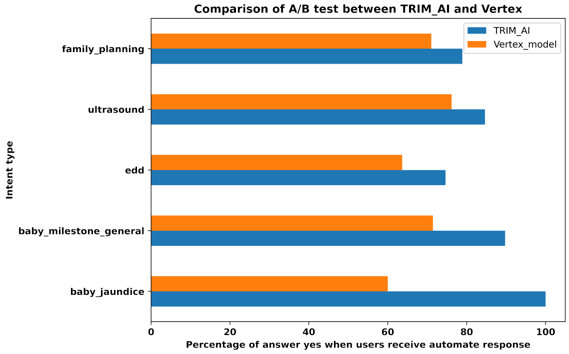

Figure 3: A/B test result based on 5 selected intents in which TRIM-AI outperforms Vertex AI most.

Conclusion

This paper proposes TRIM-AI, an NLP-based framework for automated assessment of a pregnant woman’s medical condition based on the contents of SMS messages sent by them to the PROMPTS platform. TRIM-AI leverages advances in multi-lingual pre-training and continual pre-training to handle code-mixed messages, and thus, it outperforms other state-of-the-art baselines by 15.4% (in terms of weighted F_1 scores). TRIM-AI has been deployed by Jacaranda Health since June 2022.

Acknowledgements

Wenbo Zhang, Hangzhi Guo, and Amulya Yadav were graciously supported by a Google Research AI for Social Good grant.

References

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dou, Z.-Y.; Barman-Adhikari, A.; Fang, F.; and Yadav, A. 2021. Harnessing Social Media to Identify Homeless Youth At-Risk of Substance Use. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17): 14748–14756.

Engelhard, M.; Copley, C.; Watson, J.; Pillay, Y.; Barron, P.; and LeFevre, A. E. 2018. Optimising mHealth helpdesk responsiveness in South Africa: towards automated message triage. *BMJ global health*, 3(Suppl 2): e000567.

Gotmare, A.; Keskar, N. S.; Xiong, C.; and Socher, R. 2018. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Health, J. 2021. Quarterly Impact Report Q1 2021.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. Doha, Qatar: Association for Computational Linguistics.

Kudo, T. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

Liu, P.; Qiu, X.; and Huang, X. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Maldonado, L. Y.; Bone, J.; Scanlon, M. L.; Anusu, G.; Chelagat, S.; Jumah, A.; Ikemeri, J. E.; Songok, J. J.; Christoffersen-Deb, A.; and Ruhl, L. J. 2020. Improving maternal, newborn and child health outcomes through a community-based women's health education program: a cluster randomised controlled trial in western Kenya. *BMJ global health*, 5(12): e003370.

Patil, S.; and Davies, P. 2014. Use of Google Translate in medical communication: evaluation of accuracy. *BMJ*, 349.

Scanlon, M. L.; Maldonado, L. Y.; Ikemeri, J. E.; Jumah, A.; Anusu, G.; Chelagat, S.; Keter, J. C.; Songok, J.; Ruhl, L. J.; and Christoffersen-Deb, A. 2021. 'It was hell in the community': a qualitative study of maternal and child health care during health care worker strikes in Kenya. *International Journal for Equity in Health*, 20(1): 1–12.

Appendix A: Complete Results for A/B Test

Additional information about intent distribution is shown in Table 5. Further, Table 4 shows the complete results about the real-world A/B test that we conducted in collaboration with Jacaranda Health. Only the intents which have been predicted with a precision score greater than 0.75 appear in this table. The column (*Answer "Yes"*) indicates the percentage of SMS queries for which the sender of that SMS query (i.e., a pregnant mother) was satisfied by the information provided through the AI model. Similarly, the column (with *Answer "No"*) presents the percentage of SMS queries for which the sender of that SMS query was not satisfied with the information provided through the AI model. The column called "textitTotal Questions" corresponds to the percentage of SMS queries (which belong to each intent) that were assigned to either TRIM-AI or Vertex AI model.

Finally, the row (named *Grand Total*) demonstrates that 51.8% of SMS queries have been assigned to TRIM-AI model while the remaining 48.2% queries have been assigned to Vertex AI model. For all queries sent to the TRIM-AI model, users think 79.51% of them have been answered in a satisfactory manner. For all queries sent to the Vertex-AI model, users think 78.89% of them have been answered in an unsatisfactory manner.

Appendix B: Further Statistical Analysis for A/B Test

Further, we report on the statistical significance of the results obtained in our A/B test. Our overall dataset contains 5,323 messages in total. As mentioned in the paper, this occurs when one intent among the predicted intent distribution crosses the 75% confidence threshold.

Across all intents, the performance improvement of TRIM-AI (over Vertex AI) is not statistically significant (we conducted a two-proportion Z-test and got a p-value greater than 0.05). However, of all the intents evaluated in the A/B test, TRIM-AI achieves statistically significant benefits over the Vertex AI model in two intents (i.e., *baby_milestone_general* and *baby_jaundice*). Importantly, *baby_jaundice* is a highly critical intent. The lack of statistical significance across all intents is because of the small scale of our pilot study (which was only run over a two week period).

Table 4: A/B tests results based on intents whose confidence scores are greater than 0.75 after AI prediction.

Intent Name	**TRIM-AI**			**Vertex AI**		
	Answer ”No”	Answer ”Yes”	Total Question	Answer ”No”	Answer ”Yes”	Total Question
baby_constipation	19.51%	80.49%	46.59%	19.15%	80.85%	53.41%
baby_general	30.49%	69.51%	59.85%	27.27%	72.73%	40.15%
baby_hiccups	0.00%	100.00%	73.68%	0.00%	100.00%	26.32%
baby_jaundice	0.00%	100.00%	44.44%	40.00%	60.00%	55.56%
baby_milestone_general	10.26%	89.74%	52.70%	28.57%	71.43%	47.30%
baby_milestone_teething	100.00%	0.00%	21.05%	20.00%	80.00%	78.95%
breastfeeding	29.00%	71.00%	60.61%	24.62%	75.38%	39.39%
edd	25.37%	74.63%	46.53%	36.36%	63.64%	53.47%
family_planning	21.10%	78.90%	50.46%	28.97%	71.03%	49.54%
fatigue	60.00%	40.00%	26.32%	28.57%	71.43%	73.68%
fetal_movement	20.93%	79.07%	58.90%	11.67%	88.33%	41.10%
linda_mama	21.05%	78.95%	40.43%	17.86%	82.14%	59.57%
medication_general	22.58%	77.42%	53.45%	7.41%	92.59%	46.55%
ok_thanks	15.35%	84.65%	62.93%	15.97%	84.03%	37.07%
pain_stomach	22.73%	77.27%	53.99%	25.33%	74.67%	46.01%
pregnancy_general	36.06%	63.94%	55.37%	24.06%	75.94%	44.63%
survey_response	17.97%	82.03%	49.27%	20.04%	79.96%	50.73%
ultrasound	15.38%	84.62%	29.21%	23.81%	76.19%	70.79%
urination_uti	17.39%	82.61%	68.66%	23.81%	76.19%	31.34%
Grand total	20.49%	79.51%	51.80%	21.11%	78.89%	48.20%

Table 5: Intent distribution over all test samples in A/B test.

Intent type	**Counts**	**Percentage**
survey_response	2823	53.03%
pregnancy_general	596	11.20%
ok_thanks	321	6.03%
family_planning	216	4.06%
breastfeeding	165	3.10%
pain_stomach	163	3.06%
fetal_movement	146	2.74%
edd	144	2.71%
baby_general	137	2.57%
urination_uti	134	2.52%
linda_mama	94	1.77%
ultrasound	89	1.67%
baby_constipation	88	1.65%
baby_milestone_general	74	1.39%
medication_general	58	1.09%
baby_hiccups	19	0.36%
fatigue	19	0.36%
baby_milestone_teething	19	0.36%
baby_jaundice	18	0.33%
Total	5323	100.00%