NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

FINE-TUNING A MULTILINGUAL LANGUAGE MODEL TO PRUNE AUTOMATED EVENT DATA

by

Seth W. Kyler

June 2023

Thesis Advisor:	Timothy C. Warren
Co-Advisor:	Mathias N. Kolsch
Second Readers:	Sean Eskew (TRAC Monterey)
	Daniel Ruiz (TRAC Monterey)

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2023	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
FINE-TUNING A MULTILINGUAL LANGUAGE MODEL TO PRUNE AUTOMATED EVENT DATA

5. FUNDING NUMBERS

6. AUTHOR(S) Seth W. Kyler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Every day, an enormous volume of written and transcribed media is produced, making it impossible for intelligence analysts to sift through it all without a large human workforce. However, multilingual language models can help intelligence analysts select media articles relevant to their problem set, even if they are written in a foreign or low resource language, by parsing out non-relevant articles. The Global Database of Events Language and Tone (GDELT) is a near real-time media database that releases new collections of open-source articles every 15 minutes, but its automated event coding often leads to a high number of false positive samples. To create an effective multilingual language model for parsing open-source articles, an accurate categorization and tagging of the open-source articles is necessary for training. This thesis fine-tunes multilingual language models to identify false positive open-source articles in the GDELT database using the automated coded and human verified open-source articles from the Integrated Crisis Early Warning System (ICEWS) as the training data. The fine-tuned multilingual language model is overlaid onto the GDELT search algorithm to prune out many of the false positive results, providing intelligence analysts with improved access to relevant open-source articles within minutes of publication, and enabling them to gather pertinent information in a more timely manner.

14. SUBJECT TERMS
Global Database of Events Language and Tone, GDELT, Integrated Crisis Early Warning System, ICEWS, natural language processing, automated event database, database pruning, multilingual language model

15. NUMBER OF PAGES
71

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

FINE-TUNING A MULTILINGUAL LANGUAGE MODEL TO PRUNE AUTOMATED EVENT DATA

Seth W. Kyler
Lieutenant, United States Navy
BA, Lafayette College, 2016

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 2023

Approved by: Timothy C. Warren
Advisor

Mathias N. Kolsch
Co-Advisor

Sean Eskew
Second Reader

Daniel Ruiz
Second Reader

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Every day, an enormous volume of written and transcribed media is produced, making it impossible for intelligence analysts to sift through it all without a large human workforce. However, multilingual language models can help intelligence analysts select media articles relevant to their problem set, even if they are written in a foreign or low resource language, by parsing out non-relevant articles. The Global Database of Events Language and Tone (GDELT) is a near real-time media database that releases new collections of open-source articles every 15 minutes, but its automated event coding often leads to a high number of false positive samples. To create an effective multilingual language model for parsing open-source articles, an accurate categorization and tagging of the open-source articles is necessary for training. This thesis fine-tunes multilingual language models to identify false positive open-source articles in the GDELT database using the automated coded and human verified open-source articles from the Integrated Crisis Early Warning System (ICEWS) as the training data. The fine-tuned multilingual language model is overlaid onto the GDELT search algorithm to prune out many of the false positive results, providing intelligence analysts with improved access to relevant open-source articles within minutes of publication, and enabling them to gather pertinent information in a more timely manner.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
GDELT	Global Database of Events Language and Tone
ICEWS	Integrated Crisis Early Warning System
LaBSE	Language Agnostic BERT Sentence Embedding
LLM	Large Language Model
MLM	Masked Language Model
MTEB	Massive Text Embedding Benchmark
NLP	Natural Language Processing
PLM	Permuted Language Modeling
RoBERTa	Robustly Optimized BERT Pretraining Approach
SetFit	Sentence Transformer Finetuning
TLM	Translation Language Modeling
XLM	Cross-lingual Language Models

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I feel extremely blessed to have had the opportunity to study at NPS and pursue a thesis topic of interest to me. My deepest gratitude goes to my advisory team: Dr. Warren, Dr. Kolsch, MAJ Eskew, and MAJ Ruiz. Thank you for guiding me through the thesis and research process, making yourselves readily available to help me work through obstacles, and overall encourage me. The knowledge I have gained under your instruction is well complimented by your positive attitudes and exemplary character.

A deep thanks also goes to my soon-to-be-wife, Lauren, who patiently listened to me discuss the many confusions and hurdles I faced and eventually overcame in the process of researching and writing this thesis. I love you. Thank you for listening.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In this age of information, locating relevant sources amidst the abundance of available media articles and transcripts has become an increasingly daunting challenge. If time is limited, the task increases in difficulty because fewer sources can be consulted. Finding relevant articles quickly is especially problematic in the intelligence community, where an analyst monitoring local events in a country of interest for indications of violence watches the local news stations and newspapers for events that reflect growing unrest. While growing unrest does not necessarily lead to violence, the intelligence analyst strives to use the information available to give the commander time to make decisions. Information from the media can be unreliable, but one method analysts use to overcome media biases is pulling from multiple sources, often aggregating the articles and transcripts in databases. Sifting through these multiple sources is a time-consuming task.

This thesis seeks to support the synthesis of information into intelligence by using multilingual language models to winnow though databases of media articles and transcripts and return the relevant information in a time-conscious manner. It develops a model to provide intelligence analysts with relevant media articles and transcripts on current and breaking events around the globe, enabling them to efficiently sort and internalize the information in front of them, and process it into intelligence to inform a commander. The setup discussed in this thesis trains a multilingual language model on data tagged and sorted from a human-coded or algorithmically winnowed event dataset to effectively prune media articles and transcripts tagged by a machine-coded, automated, event dataset. An objective of this research is to provide a proof of concept for future artificial intelligence pruning techniques to support DOD and civilian intelligence analysts. The scope is limited to exploring multilingual language models with the intent to fine-tune those models to enable efficient data pruning as relates to open-source collecting and querying of protest events.

The evidence presented here demonstrates that multilingual language models can effectively prune many of the false positive samples out of a large database such as

GDELT. In our first approach we implement a full training loop, in the second approach, a few-shot learner. While the few-shot learner is ideal in a low-resource environment, the full training loop yields better results. The LaBSE model shows particular promise, achieving the highest overall accuracy rate, highest F1 score, and highest ROC-AUC, demonstrating accurate classification of over 72% of the total records.

A. KEY TERMS AND DEFINITIONS

Since the meaning of words varies based on context, below are the definitions of certain words as they are used in the context of this thesis.

- Protest event: an incident in which the actors strongly express their opinions through demonstrations, picketing, rioting, or another form of public display that publicizes their discontent.
- Multilingual language model: a collection of computer algorithms that receive as inputs words and sentences of at least two languages, and processes those words and sentences in a similar manner to how a human would process those words and sentences.
- Fine-tune: Training a pre-trained language model on new data to improve its performance on a specific task.
- False positive: instances labeled as protest events by the classifier that are not actually protest events [1].
- False negative: instances labeled as not protest events by the classifier that are actually protest events [1].
- True positive: instances labeled as protest events by the classifier that are actually protest events [1].

B. OPERATIONAL NEED

Every day, thousands of media articles and transcripts are produced and pulled into databases where analysts can access them. Because of the sheer volume of written

material, analysts cannot view every article or transcript as it becomes available. Rather, they enter key words as search criteria and rely on a search engine to return relevant material. Search engines sifting through databases often rely on lexical search, looking for key words spelled exactly the way the user entered them. By ignoring even slight differences in spelling and grammar, however, the lexical search may not return relevant information simply because it did not exactly match the search criteria. Due to the many writing styles, spellings (color vs. colour, humor vs. humour), and articles written in foreign languages and translated to English, the lexical search can easily ignore otherwise relevant information. A semantic search, which seeks to match the searcher's meaning through context, uses content and intent as the criteria. This method is characteristically more difficult because content and intent rely on context and an understanding of the communication used. Natural language processing (NLP) leverages these search methodologies, and others, by training machines to process and respond to information in a way similar to the manner in which humans process and respond to information. A subset of NLP, multilingual language models, ingest and evaluate information from a variety of languages and can be used to find similarities in content, despite the publication language.

Media aggregation organizations commonly use machine processes to collect their extensive datasets in a timely manner, often sacrificing some accuracy for the production speed that comes with automation. How the datasets parse through collected data usually directly correlates to time. Some datasets, such as the Armed Conflict Location & Event Data Project or the Integrated Crisis Early Warning System (ICEWS), use human coders, or a combination of automated winnowing and human coders, to process the data before publication. [2], [3]. Human coding takes time, which delays the release of information by at least a week, in most cases. Other datasets, such as the Global Database of Event Language and Tone (GDELT), rely solely on automation to process the data. While automation results in a much quicker release of information, it also comes with media articles and transcripts that discuss information irrelevant to the event queried, hereafter known as false positive returns. False positive returns force the analyst to manually sift through the media articles and transcripts to find the relevant

among the irrelevant, which takes time. The analyst may also ignore the dataset altogether, but that may lead to the omission of indicators pertinent to the problem set.

GDELT relies solely on automation to process the data pulled from thousands of news sources and pushes out a new dataset every 15 minutes. Unfortunately this process leads to a high number of false positive returns [4], [5]. As the fastest release of globally inclusive automated information to date, GDELT can provide valuable information in a timely manner to the intelligence analyst because of the sheer volume of data available on a near-real time basis. The downside is that the coding process used can lead to inaccurate—false positive or false negative—tags, so search results may have extraneous media articles or transcripts, often described as "noise" [6]. Since the automated coding algorithm has not been made publicly available, users cannot review the tagging process. This lack of accessibility leaves them with a hypothetical black box that takes an input—the user's search criteria—and produces an output—the search results—without the user having the ability to fine-tune their search criteria based on an examination of the inner workings of that black box. These circumstances make GDELT a potentially valuable supporting resource to expand on data provided by a more accurately coded source, but too "noisy" to serve as a reliable primary resource [6]. With an automated algorithm that has learned from human-coded or algorithmically winnowed datasets, however, the false positive shortcomings of the automated coding process may be overcome. A trained automated algorithm may reduce the number of irrelevant results returned, increasing the fidelity of the near real-time dataset.

This thesis seeks to fine-tune a multilingual language model to prune out the false positive media articles and transcripts, despite their tagged codes fitting the search criteria. The goal is for intelligence analysts to receive a higher volume of relevant articles and transcripts in response to their search queries. The breadth and timeliness of the GDELT database makes it a powerful resource worth the effort to reduce the level of "noise" in the returned search results.

C. RESEARCH QUESTIONS

In this thesis we ask, how can we fine-tune a multilingual language model so that queries of the GDELT database return URLs linked to an actual protest event at a higher rate than the existing system? Previous research has estimated the current return rate of true positives in GDELT at 21% [7]. We will train the multilingual language model on another database, ICEWS, that also covers protest events [3].

Instead of reinventing how GDELT identifies and subsequently labels a protest event, the multilingual language model will demonstrate how to leverage other resources to filter out extraneous results. In the process, we will ask the following questions:

1. What are some possible multilingual language models on which to train a human-coded or algorithmically winnowed data set?
2. Which of these multilingual language models is accessible for use in research?
3. What steps are necessary to train a multilingual language model on data from human-annotated or algorithmically winnowed open-source event databases?
4. How can a multilingual language model be fine-tuned to prune the automated event data accurately?
5. How can the accuracy of true positive results returned be measured?
6. What is the tagged accuracy rate of protest event data currently pulled through GDELT?
7. What is the tagged accuracy rate of protest event data pulled through GDELT and pruned using a fine-tuned version of a multilingual language model?

D. THESIS ORGANIZATION

This thesis consists of five chapters. The first chapter outlines the problem of filtering through an abundance of media articles and transcripts and proposes a solution.

Chapter II provides background information on the databases used and descriptions of the multilingual language models applied in this research. In Chapter III, the methodology of pulling information from the selected databases and the process of fine-tuning the multilingual language models is covered. Chapter IV weighs the results of each model and discusses if the proposed approach achieves its stipulated objectives. It also lays out the strengths and weaknesses of the multilingual language models used for winnowing through databases of media articles and transcripts. Chapter V explores the implications of the results and provides avenues for future work in this area. Our code is available at https://github.com/sk8412/GDELT.

II. LITERATURE REVIEW

Knowing from where databases pull news articles and transcripts, and understanding how they categorize and organize these media pieces leads to a knowledge of how these databases can be leveraged to make parsing through articles and transcripts less time-consuming. Understanding how multilingual language models work is also essential because each multilingual language model has different strengths and weaknesses. Some will more effectively sift through media articles and transcripts than others. For what follows in this chapter, we discuss the databases used, the shortcomings of GDELT that spurred this research, and the multilingual language models tested.

A. OPEN-SOURCE DATABASES

Multiple entities compile datasets of events gleaned from media articles and transcripts, with reputations for accuracy and timeliness varying. For the application of this thesis, accuracy matters over timeliness, so ICEWS was chosen as the training dataset because of the scholastic work supporting its accuracy in distinguishing actual events from irrelevant events [6]. It, like GDELT, uses the CAMEO codes to categorize events, creating a natural alignment for using ICEWS to prune GDELT. The power behind these two databases is they are accessible to anyone—including researchers, analysts, and journalists—via the internet, providing a broad audience with pertinent information, provided they can find that pertinent information among the data. Additionally, both databases record events beyond protest events, so they have broader future applicability across diverse problem areas.

1. Integrated Crisis Early Warning System

Hosted by Lockheed Martin, ICEWS combines multiple statistical models to achieve an advertised event coded accuracy of more than 80% [3]. It pulls media articles from over 100 data sources and 250 newsfeeds, processing the articles using shallow-parsing Jabari and BBN's Serif NLP technology [3]. Both Jabari and BBN's Serif are NLP technologies used to code and sift the events in the media articles and transcripts analyzed. The precursor to BBN's Serif NLP used for event coding in ICEWS, Jabari

NLP, consistently scores below BBN's NLP technologies in terms of coding precision [8], [9]. Its continued use serves as an additional filter, despite BBN's superior NLP technology. BBN's Serif NLP technology analyzes the text and pulls structured information, such an entities, relationships, and events [9]. The published information is made available through a repository called Dataverse which holds the categorized events, each stamped with an event geolocation and date [10].

ICEWS has undergone evaluation tests, and its categorization, when compared to that of trained human coders, ranges between 74% and 86% in the three categories of protest, coercion, and violence [8]. Beyond using automated techniques to process the data, ICEWS also uses a variety of models to make predictions on crises, including rebellions, insurgencies, and ethnic and religious violence [8]. By averaging the predictions together using an ensemble method, the resulting model has shown fewer false positive samples or false negative samples than any of the individual models [8]. While it relies on NLP technology and does miss some events, ICEWS provides more certainty on the coding of events compared to GDELT [8].

2. Global Database of Events, Language, and Tone

As previously mentioned, the GDELT database provides access to an enormous amount of information. It "monitors the world's news media from nearly every corner of every country in print, broadcast, and web formats, in over 100 languages," codifying events in over 300 categories [4]. Using a "realtime streaming news machine translation," 98.4% of the non-English monitored media volume is translated into English and categorized [4]. Each event entry includes geographic location coordinates, although closer analysis of the locations provided indicates that GDELT may overstate the geographical spread of events [8]. Such a trend makes sense given the wide aperture through which GDELT pulls information. Comparison showed it does provide more specific locations than ICEWS, which tends to cluster events in population centers [8].

Perhaps most significantly in terms of real-time reporting, the GDELT database updates with new information every 15 minutes [4]. This feature gives it incredible potential for use as an aggregate source for ongoing or trending events or incidents. Due

to the incredibly large amount of data processed by GDELT every day, it uses powerful deep-learning algorithms instead of human coders to process the information [4]. There are no human coders to verify the article or transcript has received the correct event code. GDELT's powerful potential stems from the extensiveness of its sources and ability to process and provide so much information in a relatively short span of time.

While GDELT will provide multiple reports on the same event, which ensures that it maximizes reporting, this capability leads to a high number of false positive reports [8]. In experimentation to test the accuracy of GDELT to correctly identify a protest event, after filtering and de-duplication of events, only 21% of the articles pulled represented an actual protest event [7]. Additionally, unlike ICEWS, GDELT does not have a method for automatically winnowing down the stories and eliminating duplicate reporting [8]. Both databases use the same automated parsing and coding system, Conflict and Mediation Event Observations (CAMEO), which breaks sentences down into three categories: SUBJECT VERB OBJECT [7]. While efficient, the CAMEO breakdown into the three categories can lose context, which can lead to duplication of reported events [7]. GDELT has not implemented a strategy to rectify this issue in the same way the creators of ICEWS have. False positives are thus the major shortcoming of GDELT. While the breadth of information categorized would appear invaluable, it must be proven reliable to be accepted as such.

While GDELT translates text for more than 65 different languages to English text before assigning classifications, translation requires an immense amount of computational power [11]. Using a multilingual language model enables the pruning of any text that GDELT does not translate. More importantly, it allows for a setup in which the media pieces are pruned prior to any translation, which saves both computational power and time as only the true positive media articles and transcripts are then translated into English. How accurately GDELT or any other translator translates media pieces is outside the scope of this thesis.

B. GDELT AND NATURAL LANGUAGE PROCESSING

GDELT allows anyone with internet access to "see the world through others' eyes…empower [ing] local populations with the information and insights they need to live safe and productive lives" by monitoring media coverage from every country in the world [4]. Its immense computational power and availability allows the military, intelligence community, researchers, and journalists to identify incidents and event trends as indicated by local reporting. The automated processing of data, which utilizes NLP technology to translate and code the media articles and transcripts, brings vast amounts of relevant and useable data within the user's reach. Human translating and coding of the entries is not feasible due to the sheer amount of data, so the use of NLP in GDELT makes what is humanly impossible a reality [1].

NLP is the science behind enabling computers to comprehend written and spoken words in a manner similar to how humans process written and spoken words. Because the meaning of words change based on their positions in a sentence, NLP must account for context. Some models accomplish this by using a technique called bi-directional encoding, a procedure in which the model recognizes a word and then learns the words placed on either side of it in the sentence. Bidirectional Encoder Representations from Transformers (BERT), developed by Google in October 2018, uses this learning technique [12]. The other technique used in BERT, transformers, propelled NLP forward and thrust BERT into prominence.

Transformer algorithms utilize multiple self-attention layers, which enable the model to focus on the most relevant parts of the input. It focuses through weighted key-value pairs and queries, which describe the specific way the model learns context [13]. As the model receives an input in the form of text, it transforms that text via tokenization to a key. The encoder, which consists of multiple self-attention layers, receives keys, values, and queries from the previous layers in the encoder [13]. This encoder section is where the model learns how token probabilities shift in response to context. Based on these probabilities, the encoder produces a vector embedding of each input text token, such that words with similar vectors are more likely to appear in similar contexts. The decoder self-attention layers analyze a sentence up to and including a word, and then

mask the rest of the sentence to the right of that word [13]. This process acts as a test for the model during the learning phrase because the model will predict the masked word in the sentence based on what it has learned from the encoder self-attention layers. It is in the encoder-decoder attention layers that the model compares what it learned against what it suggested, and subsequently weights the values assigned to keys in accordance with how accurately it assesses its suggestion [13]. These key-value pairs are fed back into the decoder self-attention layers so that the model can receive feedback to improve its suggestions. The models iterate through massive text corpora to organically learn complex contextual dependencies, a process that is both costly and time intensive. Once a model has achieved a baseline proficiency from this initial learning, however, it can be applied to general tasks and be expected to perform relatively well.

Later iterations of learning, called fine-tuning, build upon the initial training by feeding the model a corpus of texts specific to a problem set. This powerful training technique allows application specific use of the model without it needing to be built from the ground up [14]. Fine-tuning usually leads to better results in that specific task than a non-fine-tuned model because it has been trained on more examples related to its assigned task [15]. Fine-tuning a model saves both time and costs because it avoids training the model from scratch, and usually leads to more accurate results.

1. Drawbacks to Using GDELT in Research

GDELT is an example of the *availability* of data not guaranteeing the *accessibility* of data, because while the data is there, it must be correctly categorized to be associated with related content [11]. As previously discussed, this obstacle is overcome by NLP, but databases reliant on machine learning in general, and NLP in particular, are prone to biases [1]. Much of this bias comes from the text corpora on which the models were trained or fine-tuned, so it is important that the training corpus be as diverse as possible.

Additionally, comparing the results of GDELT against similar databases, such as ICEWS, has shown that GDELT includes a higher number of false positives in its search returns [1]. Prior work using machine learning classifiers such as logistic regression, naïve Bayes, and Support Vector Machines, suggest that the higher rate of false positive

labels are not systematic [1]. With no definable pattern on which to base false positive judgements, Hoffmann, Santos, Neumayer, and Mercea resorted to human coding. Their approach still required sifting through a large amount of data to find the relevant information, again harkening back to the issue of GDELT data not necessarily being accessible. While an attainable operation in academia, a similar operation is unsustainable in a military or intelligence environment where new data is constantly being received. GDELT analyzes and classifies links to media articles and transcripts every 15 minutes, adding 70,000 to 160,000 entries every 24-hour period, so the required human workforce to process such an amount of data on an ongoing basis would be enormous.

While the true cause of GDELT mislabeling events cannot yet be determined, two trends have emerged from the analysis of protest events, which is the focus of this thesis. First, many entries labeled as protest events contained no reference to protests [1]. Second, the automated geo-classification used by GDELT caused some of the mislabels [1]. Specific to this thesis and its focus on protest events, a false positive in a GDELT entry means the event happened in a different place than reported by GDELT, or did not happen at all [1].

Despite the known issue of false positive samples returned when using GDELT, researchers tend to still use the data in its raw, unfiltered form [1], [6], [16], [17]. The use of the database underscores the resources researchers and analysts lack to monitor news in real-time, so they have little option but to utilize GDELT and similar databases, accepting the biases and false positives associated with the database's selection and processing of data, but not necessarily taking steps to winnow out the irrelevant information [1].

2. Related Research in GDELT and Multilingual Language Models

Compared to ICEWS, and as just one indication of its propensity to return false positives, GDELT reports a larger number of protests in various countries and across various types of actions than ICEWS [1]. Since GDELT includes only a URL in its return, despite the classification being based on the actual story, identifying the false

positive samples becomes particularly problematic because the user cannot simply scan the content of the media pieces and rapidly eliminate the false positive samples [1]. Depending on the number of results returned, the user may not have the time to scan anyway. Thus, a faster, and more reliable method for identifying the false positives must be implemented. This research implemented a method to follow the URLs and pull the linked titles and articles, which were then analyzed against events during the same time period pulled from ICEWS, providing an automated means of distinguishing true positive from false positive events without requiring a user to manually view each story.

Previous work by Wiriyathammabhum pitted three language models—XLM-RoBERTa-base, mLUKE-base, and XLM-RoBERTa-large—against each other and based on the consistent superior performance of XLM-RoBERTa-large over the smaller language models, concluded "language model capacity matters a lot for multilingual tasks" [18]. While a smaller multilingual language model does not typically perform as satisfactory as an multilingual LLM, adding entity knowledge can make the small multilingual language model perform more satisfactory, although typically at a cost [18]. Given the computational and monetary costs associated with an LLM, our research focused on implementing smaller multilingual language models.

Tests conducted by Connearu et al. showed that a single LLM can work for all languages, without suffering per-language performance [19]. Performance for low-resource languages, those languages in which less published content is available, often suffer because language models have less available data on which to train. To overcome lack of training data for a low-resource language, the model can be trained on other low-resource languages, which results in more satisfactory cross-lingual performance, at least up to a point [19]. Referred to as the curse of multilinguality, the tradeoff between increasing the number of low-resource languages up to the point when performance actually degrades can be overcome by increasing the language model capacity [19]. Thus, training the model on more and more low-resource languages to overcome the lack of data for a single low-resource language works only up to a certain point. Finding adequate training data in other low-resource languages can also be problematic as the languages are labeled low-resource languages for a reason.

Regarding multilingual language models, Wiriyathammabhum demonstrated through scatter plots that similar articles printed in different languages show the protest classification of one document inseparable from the protest classification of documents written in other languages. The data from each language is plotted in regions based on the classification of the data, not on the language of the document [20]. Furthermore, fine-tuned multilingual language models place the information contained in the text in the same space, regardless of language [20]. The implication is that multilingual language models can correlate similarities in the meaning of the text, even if the text is written in different languages.

C. LANGUAGE MODELS

Language models come in many sizes and with varying strengths based on the data on which they were trained and the mechanisms by which they process the data. This thesis initially considered using LLMs because they typically result in more accurate results, but an issue to overcome right away is having enough memory to store and run an LLM. Some LLMs contain billions of parameters, which require an enormous amount of memory, and must run using GPUs vice CPUs. Providers of such LLMs, like GPT-3 or GPT-J, allow customers to access their servers remotely, negating the need to download the LLM onto a local machine. Use of these LLMs comes at a cost, however, with prices ranging from $0.0004 to $0.0200 per 1000 tokens, depending on the level—how many parameters are utilized—purchased [21], [22]. A less costly solution, and therefore one that can be more easily implemented by analysts and researchers, is to take a smaller language model and fine-tune it to identify articles reporting protest events.

We chose three multilingual language models, running them first individually, and then in combination with SetFit, a Sentence Transformer few-shot fine-tuning framework. This process allows us to not only compare the performance of the multilingual language models against each other, but also their performance fine-tuned on few-shot learning against their performance as a standalone model. The first model tested was Language-Agnostic BERT Sentence Embedding, a multilingual language model tested by Mendieta in the prior work on which this thesis is built, and listed on the

Massive Text Embedding Benchmark (MTEB) Leaderboard [23], [24]. (The MTEB Leaderboard provides comparison of text embedding models on a variety of embedding tasks and lists those that are achieving the best success to date.) The second model is Paraphrase-multilingual-MPNet (multi-MPNet), also listed on the MTEB Leaderboard. It is a hybrid model of BERT and cross-lingual language models (XLM), which paved the way for cross-lingual training by combining masked language models (MLM) and auxiliary position information in its predictions [25]. Finally, we test XLM-RoBERTa, a hybrid model between XLM and Robustly Optimized BERT Pretraining Approach (RoBERTa), Facebook's version of BERT [19].

1. Language-Agnostic BERT Sentence Embedding

LaBSE supports 109 languages and incorporates both pre-training and fine-tuning strategies. As a derivative of BERT, it utilizes the bidirectional dual-encoder transformer model, which enables the model to learn context about a word from looking at the words placed on either side of it in the sentence. The dual-encoder aspect that it gleans from BERT means the two encoders encode source and target sentences separately, and this method has proven effective for cross-lingual language embeddings [26]. These sentences are evaluated by the cosine similarity parameter to determine their similarity and the result represented through a dot-product scoring function. As a multilingual embedding model, LaBSE maps text into a shared vector space, also referred to as an embedding space. Similar words, regardless of language, will lie closer to each other in the embedding space than unrelated words.

LaBSE combines the MLM and translation language model (TLM) techniques during pretraining to achieve a translation ranking protocol, which is made possible by use of the bi-directional dual encoders [26]. MLM uses a mask token to cover a word chosen at random in a sentence, and the model then tries to predict what that word should be. The model's guess is compared to the training data, and the model receives feedback from its guess, which it then processes for future word predictions. TLM enables this process to occur in a multilingual setting by including translated sentence pairs concatenated together [26]. Compared to other multilingual language models produced at

the time, LaBSE outperformed prior work done in English, French, Russian, and Chinese, and did it all in a single model [26]. Previous models were generally limited to bilingual capability. Furthermore, LaBSE performs well on languages for which it does not have training data, likely due to similarity of vector placement in the embedding space [26]. Thus, LaBSE proves helpful when analyzing low-resource languages because training data is not required to work with the media articles and transcripts in that language. While the MTEB is constantly updated, as of June 2, 2023, LaBSE is ranked 38 out of 75 [24].

2. Paraphrase-Multilingual-MPNet

Multi-MPNet, the multilingual version of paraphrase-MPNet, received training in 50+ languages to become a multilingual language model [27]. The creators of Paraphrase-MPNet took another pre-trained model, MPNet by Microsoft, and fine-tuned it on an additional one billion sentence pairs that focused the training on paraphrase detection and similarity [27]. Tracing the evolution of multi-MPNet starts with MPNet.

MPNet is a hybrid model that uses the MLM training method found in BERT and the permuted language modeling (PLM) found in XLNet, which uses an autoregressive method to predict the tokens in a sequence [25]. Song et al. found that by combining MLM and PLM in a hybrid model it can both grasp the position of each word in a sentence and learn the semantic relationships among the predicted tokens [25]. An assumption of MLM is that the masked tokens are independent, so the predictions are made without any relationship to each other [25]. PLM takes the opposite approach and treats the predictions as dependent on one another, allowing for a deeper semantic understanding [25]. The MLM technique works well for learning the position of each word in relation to the others, and the PLM technique works well for incorporating the semantic relationships. Combining MLM and PLM enables MPNet to leverage the most information while predicting the tokens under the mask by taking into consideration how many tokens are covered [25]. As it predicts the token under each mask, the model then takes the predicted token into consideration in predicting the subsequent tokens in the sentence [25].

To develop paraphrase-MPNet, the researchers took datasets containing data at the sentence level and implemented a self-supervised contrastive leaning approach [27]. Contrastive learning at the sentence level takes a sentence pair, gives the language model one of the sentences, and forces it to select the other sentence in the pair from a pool of randomly selected sentences [27]. Given an input sequence, paraphrase-MPNet produces a representative vector that captures the semantic information [27]. Paraphrase-MPNet maps sentences to a 768-dimensional vector space and is best used for information retrieval, clustering, or sentence similarity tasks [27]. To make multi-MPNet, paraphrase-MPNet was trained on parallel information in 50+ languages [27]. As of June 2, 2023 multi-MPNet is ranked 29 out of 75 on the MTEB [24].

3. XLM-RoBERTa

XLM-R, a multilingual MLM pre-trained on text in 100 languages by Facebook, has a vocabulary size of 250,002 tokens and has received training on 2.5TB of CommonCrawl and Wikipedia data [19], [28]. It uses the concept of cross-lingual transfer learning from XLM, which means the model is trained in English for a particular purpose, and then used to solve that task in a different language [28]. In the case of XLM-R, it learns from the other languages in which the training data is written, allowing it to perform particularly well on low-resource languages [19]. Model testing showed that the accuracy for low-resource languages improves the most when it receives training on all languages, not just English or the test language [19].

Combined with XLM is RoBERTa, yet another of the many derivative models from BERT. RoBERTa is more robustly trained than BERT, receiving training on a vocabulary of 50,000 tokens compared to the vocabulary of 30,000 tokens on which BERT received training [29]. In addition to training on more data, RoBERTa also received a greater duration of training time, was fed data in larger batches, had the next sentence prediction portion removed, used longer sequences during training, and mixed up the masking pattern [29]. These changes result in both an increase in the number of parameters and an increase in performance compared to BERT.

Trained in a self-supervised fashion, meaning pretraining occurred only on text with no human labeling, XLM-R automatically processes the inputs and generates embeddings from those training texts. The MLM portion means 15% of the words in the input were randomly masked by the model, which forces it to predict the masked tokens. The model uses sentence piece tokenizer for tokenization, meaning it can flexibly form sub-word tokens by using recurring byte patterns to split the tokens [28].

4. Sentence Transformer Finetuning

A framework through which a language model can be executed and trained, SetFit advertises that it allows models to be fine-tuned with as few as eight labeled examples, making it more computationally efficient than the typical model training routine, and with comparable results [30]. This technique of training on just a few examples of labeled data, called few-shot learning, can cut down drastically on the amount of time and resources the model takes to train during the fine-tuning process. For resource-constrained environments, say with only one GPU or even just a CPU, this may be a promising approach. The use of SetFit does limit the model selected to those compatible with the Sentence Transformers library, but this is hardly a limitation given the size of this Python library.

SetFit utilizes a combination of two transformer models to identify the semantic relationship between sentences, minimizing the distance between sentences of semantic similarity and maximizing the distance between sentences of semantic dissimilarities [30]. The actual training process consists of two steps. The first compares and contrasts the sentences in pairs, creating representative vectors. The second training step takes those representative vectors and, as sentence embeddings, trains the classification head [30]. Its accuracy is comparable to that of larger models, so its real strength lies in requiring less resources to achieve similar results [30].

Because SetFit trains on vectors representative of the sentence similarities, it can be applied to multilingual language models just as effectively, although its initial testing was limited to English, Japanese, German, French, Spanish, and Chinese [30]. To assess multilingual effectiveness, Tunstall et al. trained a model via SetFit in three different

styles. First, they used training data in the same language as the evaluation data [30]. Second, they used training data in English but evaluated on a different language [30]. Finally, they used training data from all six languages but evaluated on just one language [30]. Interestingly, SetFit returned the most robust results when the model received training on English data but was then applied to another language [30].

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

The overall goal of this thesis is two-fold. The first goal is to show that a multilingual language model can reduce the number of false positive articles returned in a search. The second goal is to provide a model that future work could build into a platform for use by intelligence analysts. Specifically for the latter purpose we created a methodology that requires minimal technical knowledge and processing of the data so that the information used closely resembles the information retrieved from GDELT and ICEWS.

Building our model requires the following steps:

1. Retrieve the information from the ICEWS and GDELT databases.
2. Follow the URLs in the GDELT file to populate each entry with the linked media article or transcript.
3. Geographically and temporally compare the entries of the ICEWS file against the entries of the GDELT file to label true positives and false positives from GDELT.
4. Process the GDELT media articles and transcripts through multilingual language models for fine-tuning, training a classifier to predict the true positive labels using multilingual article texts.

A. BUILDING THE DATASETS

ICEWS uploads new data on their website weekly and makes available files of data aggregated from previous calendar years. We selected the file for 2022 because it is the most recent full calendar year. While GDELT also makes all its data available for download, it is uploaded each day in 15-minute chunks, so downloading all data for 2022 is not as straightforward as it is for ICEWS. We choose to limit the GDELT data pulled to the year 2022 to match the inclusive dates of the ICEWS data.

Instead of downloading all information from 2022 available on GDELT, we built a Python script that iterates through the files. For each file, the script sifts through the

data, only keeping those entries that are coded as category 14 events under the CAMEO system. CAMEO category 14 delineates events classified as protest events, and since both ICEWS and GDELT use CAMEO classification, the same events should populate in each database.

The information imported from GDELT includes the date, actors, geographic location of the event and actors, and the URL to the source article, among other details. It does not include the title or full text of the article from which the date, actors, and geographic location are drawn. However, the proprietary GDELT algorithms read the title and full text of the article, from which it extracts its output details (Table 1). As previously alluded to in Chapter II, the *how* of GDELT pulling these output details from the title and full text is one of the reasons GDELT returns false positive samples. Some of these issues are known, and as the GDELT Event Codebook explains, are errors due to order of precedence. The TABARI ACTORS dictionary contains entries for people, organizations, locations, aliases, and geographic coordinates for use in tasks such as entity recognition, event extraction, and network analysis.

> All attributes in this [Actor Attributes] section other than CountryCode are derived from the TABARI ACTORS dictionary and are NOT supplemented from information in the text. Thus, if the text refers to a group as "Radicalized terrorists," but the TABARI ACTORS dictionary labels that group as "Insurgents," the latter label will be used . . . the CountryCode field reflects a combination of information from the TABARI ACTORS dictionary and text, with the ACTORS dictionary taking precedence, and thus if the text refers to "French Assistant Minister Smith was in Moscow," the CountryCode field will list France in the CountryCode field, while the geographic fields discussed at the end of this manual may list Moscow as his/her location. [31]

The issue of false positive samples lies with how GDELT categorizes information. As noted in the GDELT Event Codebook, an entry may list France as the location when in fact the event occurred in Russia. This shortcoming raises two important issues we seek to address using multilingual language models. First, analyzing the full text is essential to understanding the significance of the event. Second, the text-based geographic fields rather than the CountryCode may provide the more accurate indication of the location of an event.

Table 1. Description of GDELT variables. Adapted from [31].

	Variable Name	Variable Type	Description
Event ID and Date Attributes	GlobalEventID	Numeric	Unique ID for each event
	Day	Numeric	Date event took place in YYYMMDD
	MonthYear	Numeric	Date event took place in YYYMM
	Year	Numeric	Date event took place in YYYY
	Fraction Date	Numeric	Date event took place with percentage of year completed
Actor Attributes	Actor1Code	Character	CAMEO code for Actor1, including geographic, class, ethnic, religious, and type classes
	Actor1Name	Character	Actual name of Actor1
	Actor1CountryCode	Character	CAMEO code of Actor1 country
	Actor1KnownGroupCode	Character	If identified, CAMEO code for IGO/NGO/rebel organization
	Actor1EthnicCode	Character	If identified, CAMEO code of ethnicity of Actor1
	Actor1Religion1Code	Character	If identified, CAMEO code for Actor1 religious affiliation
	Actor1Religion2Code	Character	If identified, CAMEO code for Actor1 second religious affiliation
	Actor1Type1Code	Character	CAMEO code for role of Actor1
	Actor1Type2Code	Character	If multiple roles, second code
	Actor1Type3Code	Character	If multiple roles, third code
Actor Attributes repeated, but for Actor2			
Event Action Attributes	IsRootEvent	Numeric	Flag for importance of event, based on first time event occurs
	EventCode	Character	CAMEO action code for action of Actor1 performed upon Actor2
	EventBasedCode	Character	CAMEO codes of related events
	EventRootCode	Character	CAMEO root code
	QuadClass	Numeric	Primary event classification
	GoldsteinScale	Numeric	Theoretical potential impact of event on stability of country
	NumMentions	Numeric	Number of mentions identified in first 15 minutes event was seen
	NumSources	Numeric	Number of sources containing at least one mention of event in first 15 minutes event was seen
	AvgTone	Numeric	Average tone of documents containing mention of event in first 15 minutes event was seen

Event Geography	Actor1Geo_Type	Numeric	Specifies level of geographic resolution: country, state, city, etc.
	Actor1Geo_Fullname	Character	Full name of matched location
	Actor1Geo_CountryCode	Character	FIPS10-4 country code
	Actor1Geo_ADM1Code	Character	FIPS10-4 country code and FIPS10-4 administrative division
	Actor1Geo_ADM2Code	Character	Global Administrative Unit Layers code
	Actor1Geo_Lat	Numeric	Centroid latitude of landmark
	Actor1Geo_Long	Numeric	Centroid longitude of landmark
	Actor1Geo_FeatureID	Character	GNS/GNIS Feature ID for location
Data Management	DATEADDED	Numeric	Date event was added to database
	SOURCEURL	Character	URL of first news report of event

Running the script through the files and extracting only protest events takes a considerable amount of time. Extracting data for the year 2022 requires parsing through 34,438 files and took over five days, resulting in a data set of 154,418 protest events (Figure 1).

Before retrieving the title and full text of the articles, a second Python script filters the articles pulled from GDELT by comparing the date, latitudes, and longitudes in the geographic fields against the date, latitudes, and longitudes in ICEWS. Because the protest events pulled from ICEWS cover only 2022, any event pulled from GDELT that is labeled with a year other than 2022 is removed. The entries in GDELT are also marked as either a true positive sample or a false positive sample based on the date of the event and associated geographic data. Any event found in both GDELT and ICEWS, listed as occurring on the same day and with geographic centroids within 70 kilometers of each other, is marked a true positive sample. If the event is in GDELT, but corresponding data is not found in ICEWS, it is marked as a false positive sample. Running this second script took just a few minutes. Figure 1 shows how the number of protest events was winnowed down, first by eliminating any that did not have an associated geographic latitude and longitude.

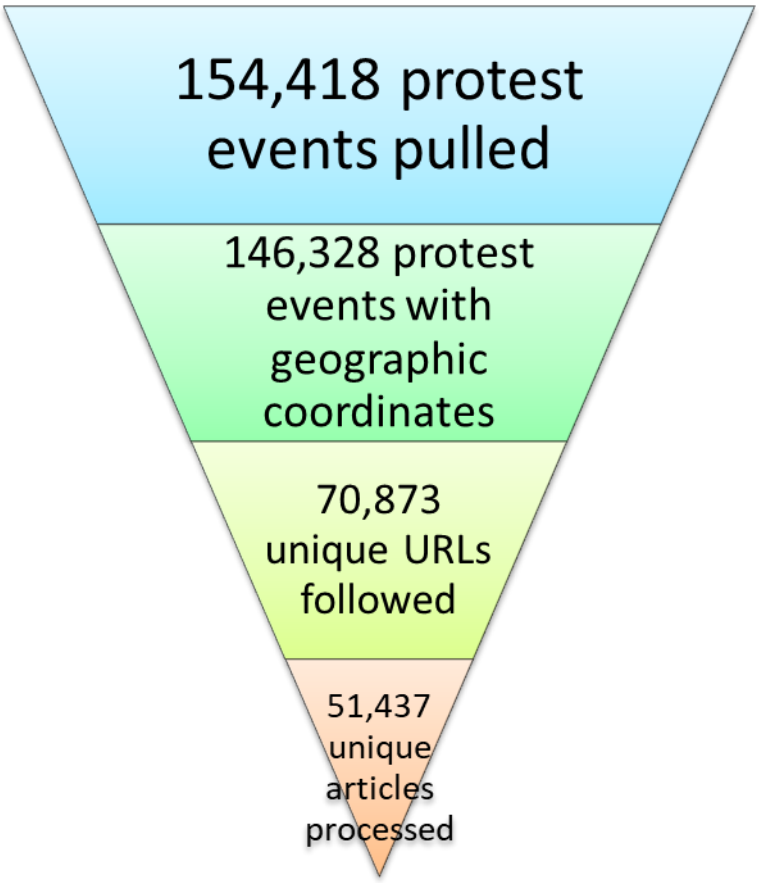


Figure 1. Winnowing of data to the final dataset

On a more technical level, the geographic comparison script creates a grid of the date, and longitude and latitude coordinates based on the ICEWS event data. The algorithm searches the grid cells neighboring the protest's location on the specified date to find any events listed in GDELT on that same day within a distance of 30, 50, or 70 kilometers. If any GDELT events are found within the ICEWS distance thresholds, the corresponding GDELT threshold column is marked as a match. These three columns, with each row marked either as a match or not a match, later serve as the labels for the multilingual language model training and evaluation.

Since the full text must be analyzed to catch the GDELT categorizing errors, a third Python script follows each URL and downloads the title and text of each linked article. If a URL no longer leads to a live page, the script notes the error and at the conclusion of running, issues a final count of how many dead URLs it found. The

resulting CSV contains seven columns: Global Event ID, Source URL, Title, Text, Match 30, Match 50, and Match 70. The Global Event ID, a unique GDELT number assigned to each listing, is used for removing duplicate events to ensure no listing appears twice in the dataset. The Source URL is from where the title and full text of the article are pulled, which are stored in the columns Title and Text, respectively. Finally, the match columns identify with a 1—match—or 0—not a match—whether a corresponding event is found in the ICEWS data within a 30-, 50-, or 70-kilometer radius, based on the latitude and longitude associated with the events. For the year 2022 the script pulls article texts from 70,873 unique URLs (Figure 1). Some of the URLs lead to the same article, so duplicate articles are removed provided the true positive or false positive flag also matches. Following the removal of duplicates and any URLs that do not return article text, 51,437 unique articles remain (Figure 1). These article texts become the primary dataset for model training and testing (Table 2).

Code for the first script—which pulls protest events from the GDELT database, and third script—which follows the article URL to download the title and full article text, draw heavily from the research of Hoffman, Santos, Neumayer, and Merceain. In the GitHub repository associated with their paper, "Lifting the Veil on the Use of Big Data News Repositories: A Documentation and Critical Discussion of a Protest Event Analysis," they made available the code they used to sift through the GDELT database and download titles and full texts of articles. We adapted this code for our purposes, making minor edits.

The final step before passing the information through a multilingual language model is to convert the data from a CSV file to an Apache Arrow Dataset. This process also splits the information into training, validation, and test sets. Running this fourth script took just a few minutes and broke the dataset down as specified in Table 2.

Table 2. Training, validation, and test dataset split with features

Dataset	Number of Entries	Features
Training	37,035	GlobalEventID SOURCEURL Title Text match30 match50 match70
Validation	9.258	
Test	5,144	

With this dataset complete, focus shifts to the culmination of this research: using the data to train, validate, and test three different multilingual language models.

B. TRAINING AND VALIDATING THE MODEL

The training, validation, and testing steps of the analysis rely heavily on the Sentence Transformers library available through Hugging Face [32]. This library hosts plugins for tokenizing, padding, implementing a model, utilizing GPUs, and evaluating the data. We built the training script with the intention of running it through multiple models, so we made use of a several automated plugins, including AutoTokenizer and AutoModelForSequenceClassification. The automated plugins select the application for the model being used, which serves well for our target audience, the intelligence analyst. Implementing the automated plugins, when available, allows for known deviations without the intelligence analyst needing to become an expert in sentence transformer architectures.

The tokenizer encodes the input texts into integer values that are fed into the language model. Based upon the model chosen, the AutoTokenizer will select one of two tokenizer methods with which to tokenize the data. The first, PreTrainedTokenizer, provides a consistent interface for all the language models, but uses more memory and takes longer to run than PreTrainedTokenizerFast [33]. Moreover, PreTrainedTokenizer can be applied to more models because PreTrainedTokenizerFast is optimized for specific model configurations. It uses Rust—a systems programming language that focuses on speed, safety, and concurrency—to speed up the tokenizing process [33]. Both

tokenization methods accomplish the goal of transforming the words or sub-words into numerical values, so the method used is not of as much importance as successfully achieving the end state.

Regardless of the tokenization method chosen, there are two options that impact the result: padding and truncation. Padding fills any unused space with a special token, usually 0s, to make all input sequences the same length. Machine-learning models usually expect input sequences to be the same length, so for sentences that fall short of the widely used 512 token limit, padding fills the unused space. On the other hand, truncation cuts an input sequence off at the token limit, whether 512 or another value. While this ensures the input sequences are not longer than 512 tokens, it also means information is lost from any longer input sequences. Thus, to optimize performance, we set padding and truncation to True. This truncates the input sequence at whatever the maximum length is for both the tokenizer and the model. The models considered here support up to a length of 512 tokens, but should a model allow a greater input sequence length, the AutoTokenizer will detect this and adjust accordingly.

Following tokenization, in which only the Text column—which contains the full article texts pulled from the webpages—is tokenized, the script removes all features except the three token-associated and label columns. Any of the Match columns can be used as the single binary label column, but we chose Match 70 because it contains the most ICEWS collaborated events. Match 70 annotates with a 1 any GDELT entry where the date exactly matches that date in ICEWS. Additionally, the latitude and longitude in GDELT must be within 70-kilometers of the latitude and longitude in ICEWS.

The three token-associated and label columns are the only information that is inputted into the model for training, validation, and evaluation. The final input data features submitted to the multilingual language models are shown in Table 3. The input_ids are the sequence of integers that represent the tokens in the input text after being tokenized. The attention_mask indicates which tokens need to be considered during the training and which are padding and should thus be ignored.

Table 3. Training features and value types

Feature	Value (dtype)
Labels	float32
input_ids	Sequence(int32)
attention_mask	Sequence(int8)

The AutoModelForSequenceClassification receives as input a model type chosen by the user and returns the specified model with hyperparameters automatically selected based on the input configuration. Once again, for the intended audience, the automatic selection will return satisfactory results without the intelligence analyst needing to extensively learn about hyperparameters settings. As discussed in Chapter II, we chose three multilingual language models based on their performance against other multilingual language models. The other necessity which eliminated many language models is the need for a multilingual capacity, necessary because not all the articles in GDELT, and thus the training data, are in English.

The batch size of a language model specifies how many input sequences are considered at a time during training, and it varies based on the model. Larger batch sizes can lead to faster training times, but also run a higher risk of overfitting. However, the risk of overfitting can be mitigated by large amounts of training data. Smaller batch sizes take a longer time to train but can result in improved performance on the validation or test data. We chose to train with as large a batch size as possible and mitigate the risk of overfitting by fine-tuning on a large training dataset and using a weight decay. By trial and error, we determined the largest batch size we can feed XLM-R and multi-MPNet is 32 input sequences and for LaBSE 16 input sequences, the limiting factor being available memory in the GPU.

The remainder of our hyperparameters, shown in Table 4, are commonly used hyperparameter values in the NLP community. In keeping with our goal of limiting the technical knowledge required to implement our model, we refrained from extensive hyperparameter adjustments.

Table 4. Hyperparameters for multilingual language models

Hyperparameter	Value
Batch Size (batch_size)	16/32
Number of Labels (num_labels)	1
Initial Learning Rate (init_lr)	5e-5
Number of epochs (num_epochs)	1
Number of Warmup Steps (num_warmup_steps)	0
Learning Rate Scheduler (lr_scheduler_type)	constant
Weight Decay (weight_decay)	0.1
Seed (seed)	42

LaBSE trained on a batch size of 16 while XLM-R and multi-MPNet trained on a batch size of 32.

We also utilize the Accelerate library, which enables the script to run across any distributed configuration—including multiple GPUs—with minimal additional code [34]. Other methods for loading the data onto GPUs or other distributed configurations exist, but they tend to be more complicated. The use of Accelerate again emphasizes our desire to keep the model setup as simple as possible while still accomplishing efficient training.

As the models chosen are pre-trained models, meaning they have already processed and learned from large amounts of data, we can improve results by fine-tuning each model on the curated data. Fine-tuning can take seconds to days depending on the amount of data input into the language model, but the length of time required to fine-tune will be significantly less than the amount of time required to initially train the model because the learning from the fine-tuning builds upon the initial learning.

To view the progress of training, we also utilize the Weights and Biases library in our analysis [35]. This allows for real time viewing of the epoch and training loss, allowing us to see when the fine-tuning has plateaued, or when overfitting might be occurring. The epoch specifies the number of times the training data is shown to the model during the fine-tuning process. During each epoch the model will repeatedly make a forward pass to generate predictions for a batch of data, calculate the training loss, and then update its parameters through backpropagation. The training loss measures the

difference between the predicted output and the actual output. The closer the generated predictions are to the actual output, the smaller the training loss value will be.

All models run on a single Tesla V100-SXM3 32GB GPU using a batch size of 16 or 32. By decreasing the batch size to four the SetFit wrapped models can be run on the free version of Google Colab or even a CPU. The training loop that trains on the full dataset uses too much memory to run on either the free version of Google Colab or a CPU, even using a batch size of four.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ANALYSIS AND DISCUSSION

This chapter discusses the implementation and results of the steps outlined in Chapter III. We evaluate two training loops, one on the full training dataset, and the second implementing the SetFit wrapper. The training loop using the SetFit wrapper completes in seconds because the model trains on a random number of eight samples from the training dataset. Eight is the minimum number of samples required to train a model using SetFit [30]. The training loop that iterates through the full dataset takes about 55-minutes training with a batch size of 16, and 35-minutes training with a batch size of 32. All three multilingual language models show a substantial improvement in true positive identification accuracy over the 21% determined by Wang et al. and our initial return rate of 19.83% [7], with the best-performing model achieving an overall accuracy rate of 72.71%.

A. DATA CHARACTERISTICS

Cleaning and preparing the data requires most of the work as the article text does not always import cleanly into the CSV. Because the language models truncate an input sequence after a set number of tokens, typically 512, the latter part of the text in a longer article is not processed by the language model. Using the tokenizers associated with each model generates estimates, for typical English text, of 100 tokens for approximately 75 words [36]. Text containing symbols vice simple characters, such as Mandarin or Arabic, generate an even higher token to word ratio. Thus, truncating the text to 512 tokens still allows for substantial portions of most articles to be captured.

Figure 2 shows that the number of protest events reported each month are evenly distributed throughout the year 2022. The exception is December, which is disproportionately low at 2%, but may reflect the annual holiday period. While it could be that protest events really did decrease during December, it may also be that the reporting of such events decreased due to reporting staff taking time off in observance of the holidays, resulting in less coverage in general. Overall, the spread of protest events

shows a relatively consistent amount of reporting throughout the year, with the deviation in December possibly due to global observances.

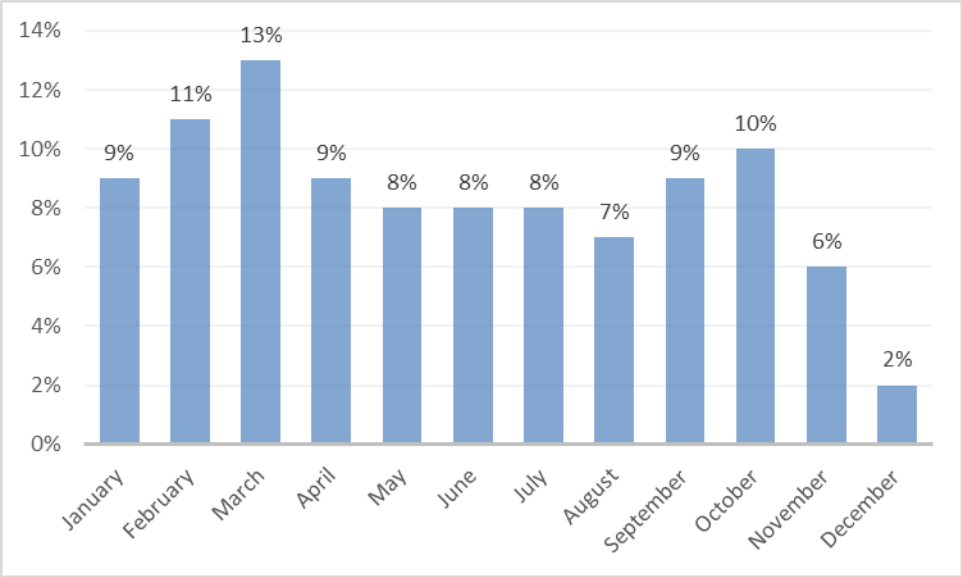


Figure 2. Percentage of protest events reported in 2022 by month

Figure 3 shows the breakdown of false positives and true positives in our dataset of 51,437 unique full text articles, as tagged by spatiotemporal matching to the ICEWS database. As discussed in Chapter II, Wang et al. manually sifted through media articles tagged as protest events by GDELT and found that only 21% were actually protest events [7]. Based on ICEWS protest event data, our dataset contains 10,201 true positive protest events out of 51,437 protest events. That is a 19.83% true positive return rate, slightly lower than the findings of Wang at al., but close enough to substantiate that GDELT has approximately an 80% false positive return rate.

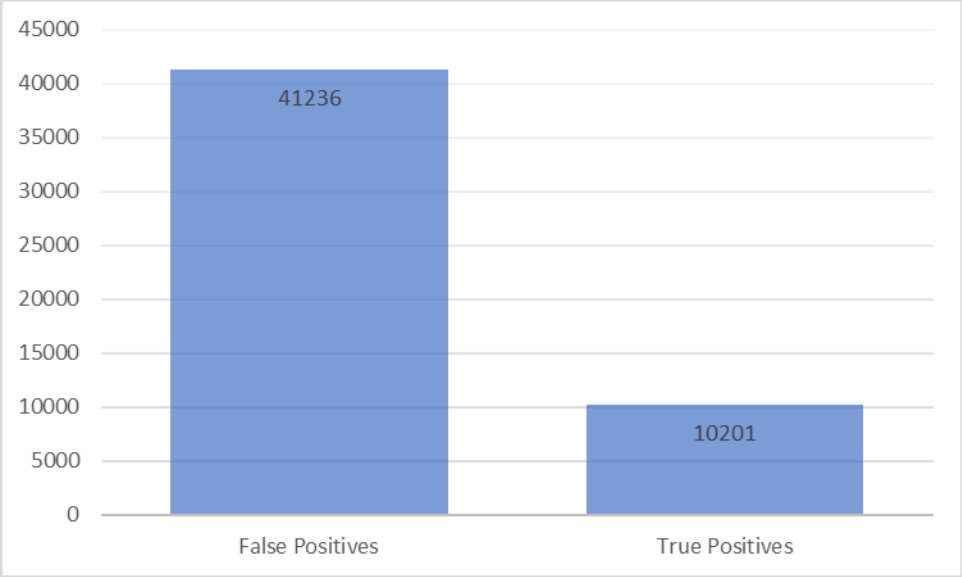


Figure 3. Count of dataset false positives vs. true positives

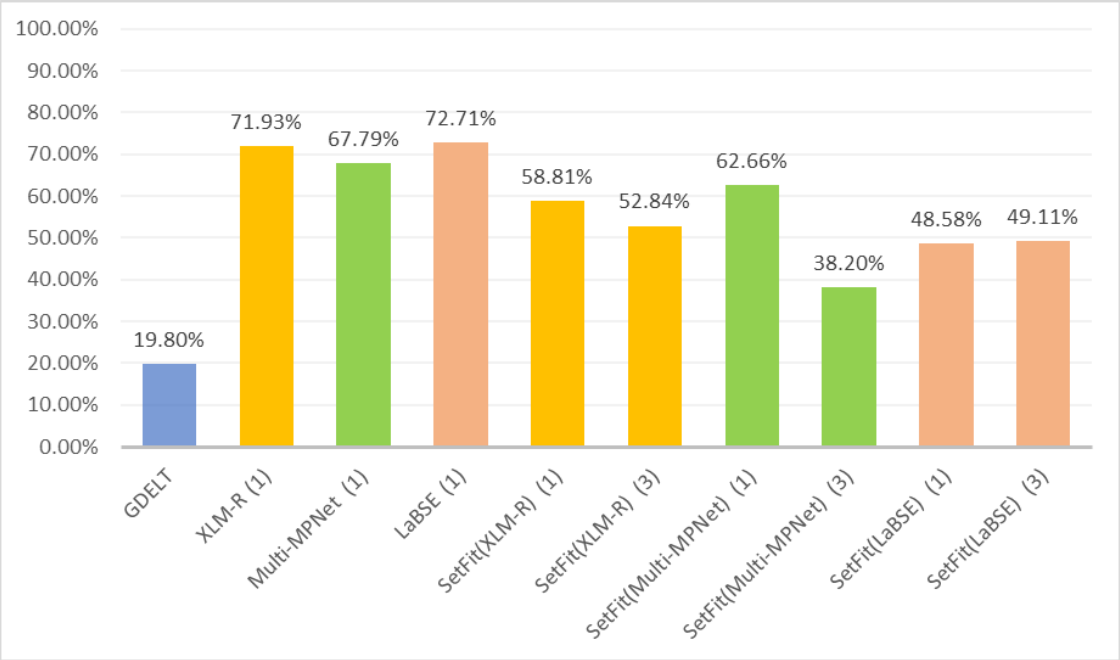
B. TRAINING LOOPS

In order to calculate model accuracy, the continuous scores generated by the model for each observation must be transformed into binary predictions, by assigning a value of 1 to scores over some threshold. Because the dataset is imbalanced, rather than relying on an arbitrary threshold of 0.5, we estimate an optimal prediction threshold for each model using the receiver operating characteristic (ROC) curve [37]. The optimal threshold is estimated by finding the threshold which maximizes the difference between the true positive rate and the false positive rate (also known as Youden's J statistic). The calculation is based on the validation data set to ensure separation from the test data set that will be used to calculate the final accuracy rates. The estimated prediction thresholds are listed in Table 5. Any scores over these thresholds are translated into positive predictions.

Table 5. Optimal prediction thresholds for models

	Optimal Prediction Threshold
XLM-R	57.28%
Multi-MPNet	55.12%
LaBSE	54.63%

After training, the multilingual language models result in accuracies ranging from 38.20% to 72.71% (Figure 4). These results are much higher than our initial GDELT return of 19.83% and show that multilingual language models can be fine-tuned to prune out false positives in the GDELT database and produce accurate results more efficiently than the original GDELT system. While the results are not perfect, the methodology provides a framework demonstrating how a multilingual language model can be applied to support the ongoing efforts of sifting through large-scale open-source data streams.



Number of epochs run are in parentheses.

Figure 4. Model accuracy results

1. Full Training Loop

After training on the full data set, the three different model architectures generate classification accuracies ranging from 67.79% to 72.71%, significantly higher than our original GDELT return of 19.83% (Table 6). Accuracy measures the fraction of correctly predicted labels, both true positive and false positive predictions [23]. Here, this refers to a binary prediction identifying whether an article is associated with an actual protest event (1), or not (0). These accuracy results indicate the models are correctly predicting

true positive and false positive samples at more than three times the rate of results originally returned by GDELT.

Precision measures the fraction of true positive samples predicted against the total number of positive predictions [23]. All three models return precision scores of 33.73% to 37.59%, indicating the models are still marking as protest events many articles that are not actually reporting protest events (Table 6). While false positive samples are still in the results, this represents a substantial improvement over the results generated without model pruning.

Recall measures the fraction of true positive samples predicted against the total number of positive instances [23]. The models return recall scores of 62.32% to 71.28%, indicating that the best model can correctly predict that an article is associated with an actual protest event more than 70% of the time (Table 6). Of all the articles associated with actual protest events in the database, however, the best model misidentifies approximately 30% as non-protest articles.

F1 score measures the harmonic mean between recall and precision [23]. It is particularly useful in datasets with large imbalances between true positive samples and false positive samples, such as our dataset, which has almost four times as many false positive samples as true positive samples. Theoretically the models could achieve a high accuracy score by predicting all samples as false positives. Because four out of five examples are indeed false positives, the returned accuracy score might be about 80%, which would lead us to believe the model is effectively pruning out the false positive samples. In fact, the models would be just labeling the samples as false positives and not identifying the true positives. F1 considers both precision and recall, providing a metric that reflects not just how many samples are correctly predicted, but rather how many samples are correctly predicted despite the majority of the samples being false positives. The models returned F1 scores of 45.80% to 47.65% (Table 6). As these values are all within 2% points of each other, this shows that the models are generally behaving similarly. Comparing across the three selected model architectures, the LaBSE model achieves the highest values for both overall accuracy and F1 score, but the differences

between architectures are small compared to the overall improvements in accuracy over the original data set.

Table 6. Accuracy, recall, precision, and F1 results

	Accuracy	Precision	Recall	F1
XLM-R	71.93%	36.30%	62.32%	45.88%
Multi-MPNet	67.79%	33.73%	71.28%	45.80%
LaBSE	72.71%	37.59%	65.07%	47.65%

ROC-AUC (the area under the receiver operating characteristic curve) concisely summarizes the rate of true positive samples against false positive samples across the full range of potential prediction thresholds [38]. ROC-AUC measures the probability that the model will rank a randomly chosen true positive sample higher than a randomly chosen true negative sample [38]. The ROC-AUC value thus measures the overall capability of the model to rank samples correctly, with 50% corresponding to a completely random classifier. Scores higher than 50% indicate the model is identifying samples at a rate better than random. All three models score over 75%, showing that our multilingual language models have a much better than random chance of correctly identifying a sample as a true positive or a false positive (Table 7).

Table 7. ROC AUC results

	ROC AUC
XLM-R	75.97%
Multi-MPNet	75.88%
LaBSE	76.73%

Increasing the training time typically leads to improved performance, however, for the sake of keeping the model simplistic and to conserve time, we trained all our models for only one epoch. We note, however, that the training loss and validation loss lines trend downward, indicating additional training would likely improve performance (Figures 5 and 6).

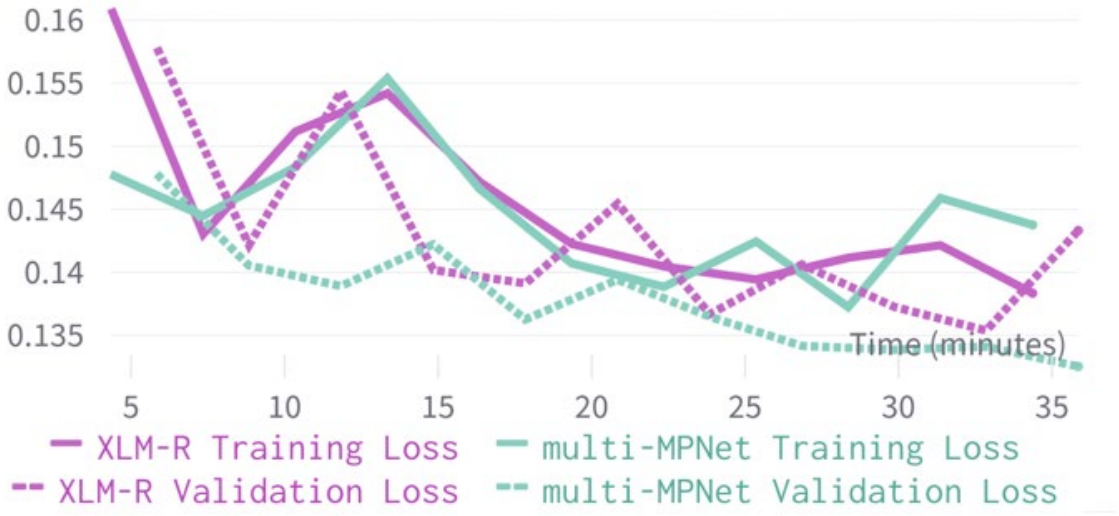


Figure 5. Training and validation loss for XLM-R and multi-MPNet

Figure 6. Training and validation loss for LaBSE

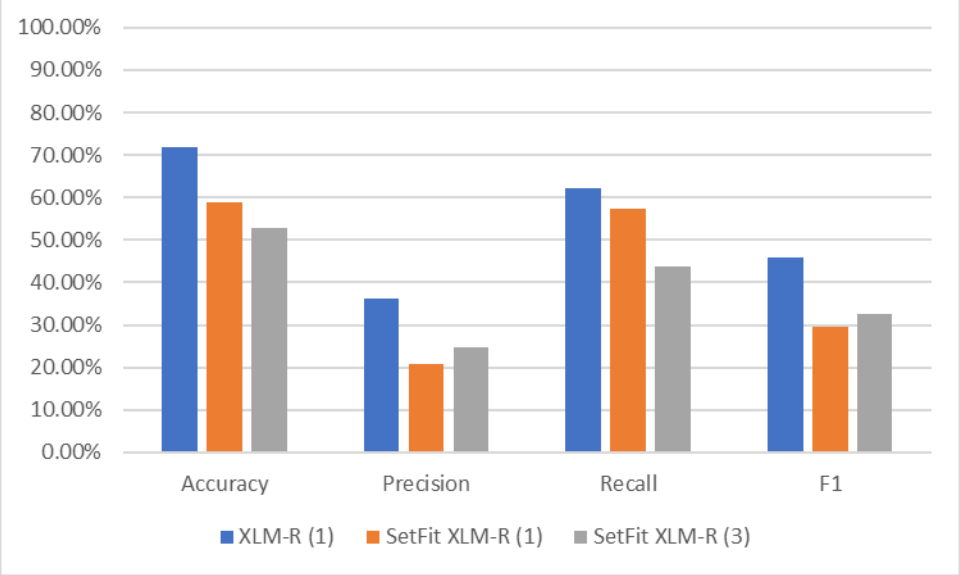
2. Few-Shot Training Loop

When using the SetFit wrapper, the models train in a matter of seconds instead of minutes (Table 8). The shorter training time required with this "few-shot" training approach is appealing because it requires far less computational resources. The metrics produced using this approach reveal the same trend seen in the full training loop, that of a better ability to correctly identify true positive samples over false positive samples.

Figures 7 through 9 show the distribution of values. The best accuracy scores from few-shot training show more than a two-fold increase over our original GDELT rate of 19.83%, which is an improvement, but not as high as the 72.71% accuracy rate achieved by the full training loop. This lower accuracy is expected, given that this approach utilizes few-shot learning. However, in resource-constrained environments, this improvement in efficiency may still be quite valuable.

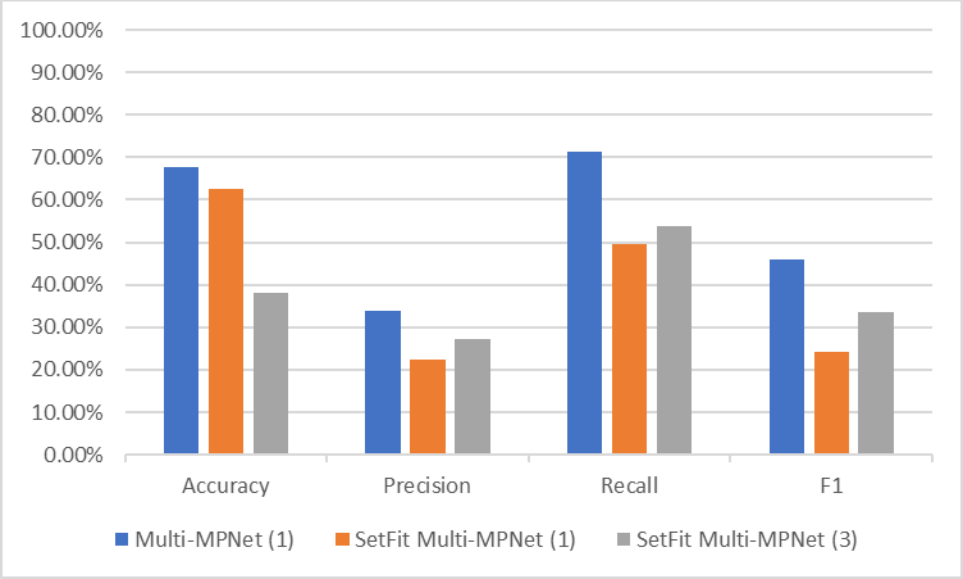
Table 8. Training time using SetFit wrapper

	Epoch = 1	**Epoch = 3**
XLM-R & Multi-MPNet	9s	27s
LaBSE	20s	59s



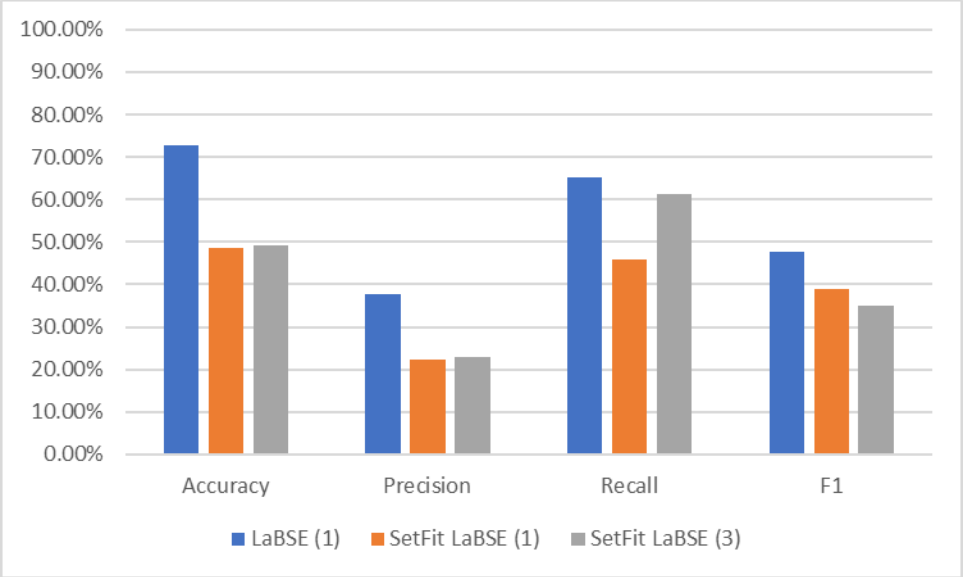
Number of epochs run are in parentheses.

Figure 7. XLM-R metrics



Number of epochs run are in parentheses.

Figure 8. Multi-MPNet metrics



Number of epochs run are in parentheses.

Figure 9. LaBSE metrics

Across the three model architectures, when utilizing the few-shot training approach, recall scores ranged from 43.79% to 61.30%, showing the multilingual language models using SetFit can improve identification of true positive samples, though

at a lower rate than when training using the full data set. Similarly, the precision ranged from 20.82% to 27.23%, showing that the models can improve the efficiency of the returned results, though again at a lower rate than achieved when training using the full data set.

Since the SetFit model runs in seconds, it was run for both one epoch and three epochs to see if the final metrics improved with two additional rounds of training. Based on the F1 score, XLM-R and multi-MPNet improve after training for three epochs, but LaBSE does not (Table 9). In contrast, when considering overall accuracy, XLM-R and multi-MPNet both show decreasing scores, whereas LaBSE shows a small improvement, when training for three epochs. This seems to indicate inconsistent effects from repeated epochs of few-shot learning, which may indicate that model over-fitting is occurring due to the smaller sample size.

Table 9. Metrics using SetFit at 1 epoch and 3 epochs

	Accuracy	Precision	Recall	F1
XLM-R (1)	58.81%	20.82%	57.43%	29.53%
XLM-R (3)	52.84%	24.77%	43.79%	32.54%
Multi-MPNet (1)	62.66%	22.40%	49.69%	24.22%
Multi-MPNet (3)	38.20%	27.23%	53.77%	33.48%
LaBSE (1)	48.58%	22.42%	45.82%	38.90%
LaBSE (3)	49.11%	22.78%	61.30%	35.00%

The number of epochs run are in parentheses.

Overall, of the three models using the SetFit wrapper, LaBSE achieves the highest F1 score of 38.9% when training for single epoch. The better performance displayed by LaBSE compared to XLM-R and multi-MPNet when training using the few-shot approach could be because LaBSE is specifically trained for cross-lingual sentence embedding. When implementing a few-shot learning technique as used in SetFit, the cross-lingual sentence embedding may help the model correctly identify samples, even though it has fewer samples from which to learn. XLM from XLM-R also uses cross-lingual sentence embedding, but RoBERTa and multi-MPNet also pretrained on a wider range of multilingual data. This broader exposure seems to work to their benefit when

there are a large number of samples on which to learn, as illustrated by all three models achieving similarly high accuracy metrics when trained using the full data set. Being reliant on a larger number of samples, however, may handicap them in few-shot learning because the models do not implement other techniques to compensate for a lower number of training samples.

The benefit of the few-shot approach is that an analyst can label just a small percentage of the data and let the SetFit wrapped model label the rest. Because SetFit can make predictions using as few as eight training samples, an analyst could manually identify at least eight samples and then implement a SetFit wrapped model. This method may work best for ad hoc winnowing of an event category where a fine-tuned model does not already exist. Ideally a model would be fully trained using data for a specific problem category, which would take more time up front, but return more accurate results.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

Ultimately our work concludes that tokenizing news articles from GDELT and training a multilingual language model is a viable method for reducing the number of false positives in the results returned when querying the GDELT database. The results show a substantial reduction in the percentage of false positive results, indicating that the effort of training such models can produce large improvements in analyst efficiency when faced with large-scale open-source data streams from online media. Future work could focus on further refining the articles retrieved, further fine-tuning the language models, and building a more user-friendly interface.

A. KEY RESULTS

Our goal is to identify the false positives returned from GDELT so they can be automatically removed, leaving an analyst with fewer false positive samples to manually code. The results presented here demonstrate that multilingual language models can winnow out false positive samples in large amounts of information. Of the model architectures considered here, the LaBSE model shows particular promise, achieving the highest overall accuracy rate, highest F1 score, and highest ROC-AUC, demonstrating accurate classification of over 72% of the total records.

We accomplish this task by generating multilingual vector embeddings for news articles, which allow for a multilingual approach because the encoded vectors capture the meaning of the thought regardless of the language in which the words are written. Vector embedding allows the language models to plot the context of the article in vector space, where it can determine vectors of similarity, allowing for a contextual comparison to other articles regardless of its written language. Using multilingual sentence transformers to compare article content is particularly powerful when analyzing articles in low-resource languages because of the inherent lack of training data for low-resource languages. The benefit of the multilingual language model is that regardless of language, many of the false positive samples and many of the true positive samples can be

automatically identified, saving analysts' time reading the returned articles and, in the case where translation in necessary, computational resources.

Our results also show that the methodology chosen accomplishes the goal of creating the foundation for a relatively simple model that could be utilized by broad audiences. The few-shot training approach implemented through the SetFit wrapper, in particular, allows the model to operate on a largely pre-existing framework that requires minimal additions or manipulations. Additionally, this approach generates results in a matter of seconds, just a fraction of the time it takes to execute the full training loop. The drawback is that few-shot training does not return quite as high an accuracy as the full training loop. Nevertheless, the improved efficiencies that can be achieved at minimal cost using this approach could be valuable in a number of settings, especially in low resource environments.

B. FUTURE WORK

Future work could more closely examine the retrieved article texts and eliminate those with generic headers or footers that have no relevance to the context of the article. Some of the article texts retrieved included terms of use clauses or copyright warnings. Although part of the original article, it is unlikely these paragraphs help the multilingual language model process the main point of the news article or transcript. Implementing an effective way to eliminate these generic clauses, and other efforts to normalize inconsistencies in the input data, may improve model performance.

In addition, the models considered here could be further optimized. Larger training data sets would likely generate more robust learning, and longer training runs utilizing more repeated epochs may further increase model accuracy. It is unclear from the results presented here whether the different model architectures would show different benefits from larger and longer training runs. It is also likely that further fine-tuning of the model hyper-parameters, such as the learning rate, could allow the models to learn more effectively.

Lastly, while our method of using multiple Python scripts and CSV files demonstrates the utility of this approach, it is hardly a user-friendly tool for analysts who

lack training in data science methods. Further work could combine these algorithmic steps and file formats into a more user-friendly software product with a graphical user interface accessible to broader audiences. We are optimistic that further development will simplify the process proposed in this thesis, making it something intelligence analysts can easily implement into their daily routines, regardless of technical background or knowledge.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] M. Hoffmann, F. Santos, C. Neumayer, and D. Mercea, "Lifting the Veil on the Use of Big Data News Repositories: A Documentation and Critical Discussion of A Protest Event Analysis," *Comm. Methods and Measures*, vol. 16, no. 4, pp. 283–302, Sep. 2022 [Online]. Available: https://doi.org/10.1080/19312458.2022.2128099

[2] ACLED, "Quick Guide to ACLED Data," March 2023 [Online]. Available: https://acleddata.com/resources/quick-guide-to-acled-data/

[3] Lockheed Martin, "Integrated Crisis Early Warning System (ICEWS)." Accessed: Sep. 30, 2022 [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/research-labs/advanced-technology-labs/icews.html#MoreInfo

[4] The GDELT Project, "The GDELT Project." Accessed: Sep. 8, 2022 [Online]. Available: https://www.gdeltproject.org/

[5] J. Hammond and N. Weidmann, "Using Machine-coded Event Data for the Micro-level Study of Political Violence," *Research and Pol.*, vol. 1, no. 2, Sep. 2014 [Online]. Available: https://doi.org/10.1177/2053168014539924

[6] M. Halkia, S. Ferri, M. Papazoglou, M.-S. Van Damme, and D. Thomakos, "Conflict Event Modeling: Research Experiment and Event Data Limitations," in *Lan. Resources and Eval. Conf.*, 2020 [Online]. Available: https://aclanthology.org/2020.aespen-1.8.pdf

[7] W. Wang, R. Kennedy, D. Lazer, and N. Ramakrishnan, "Growing Pains for Global Monitoring of Societal Events," *Sci.*, vol. 353, no. 6307, pp. 1502–1503, Sep. 2016 [Online]. Available: https://doi.org/10.1126/science.aaf6758

[8] M. Ward, A. Beger, J. Cutler, M. Dickenson, C. Dorff, and B. Radford, "Comparing GDELT and ICEWS event data," *Ana.*, vol. 21, pp. 267–297, Oct. 2013 [Online]. Available: https://www.researchgate.net/publication/303211430_Comparing_GDELT_and_ICEWS_event_data

[9] Raytheon BBN Technologies, "BBN ACCENT Event Coding Evaluation," August 28, 2015 [Online]. Available: https://www.raytheonintelligenceandspace.com/news

[10] Harvard Dataverse, "ICEWS Coded Event Data." Accessed: Jun 1, 2023 [Online]. Available: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/28075

[11] F. R. Hopp, J. Schaffer, J. T. Fisher, and R. Weber, "iCoRe: The GDELT Interface for the Advancement of Communication Research," *Comp. Comm. Research*, vol. 1, no. 1, pp. 13–44, Oct. 2019 [Online]. Available: https://doi.org/10.5117/CCR2019.1.002.HOPP

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL,* pp. 4171–4186, May 2019 [Online]. Available: https://doi.org/10.48550/arXiv.1810.04805

[13] A. Vaswani *et al.*, "Attention is All You Need," in *Adv. in Neural Info. Proc. Sys.*, 2017 [Online]. Available: https://doi.org/10.48550/ARXIV.1706.03762

[14] Hugging Face, "Fine-tune a pretrained model." Accessed: June 2, 2023 [Online]. Available: https://huggingface.co/docs/transformers/v4.18.0/en/training

[15] OpenAI, "Fine-tuning." Accessed: June 2, 2023 [Online]. Available: https://platform.openai.com/docs/guides/fine-tuning

[16] F. Bolivar *et al.,* "Understanding the sustainability framework using Big Data," presented at BBVA Research, Online, Apr. 29, 2021 [Online]. Available: https://www.bbvaresearch.com/en/publicaciones/global-understanding-the-sustainability-framework-using-big-data/

[17] D. Christensen and F. Garfias, "Can You Hear Me Now? How Communication Technology Affects Protest and Repression," *Quart. Journal of Pol. Sci.*, vol. 13, no. 1, pp. 89–117, Mar. 2018 [Online]. Available: https://www.nowpublishers.com/article/Details/QJPS-16129

[18] P. Wiriyathammabhum, "ClassBases at CASE-2022 Multilingual Protest Event Detection Tasks: Multilingual Protest News Detection and Automatically Replicating Manually Creatalkiaed Event Datasets," *arXiv*, Jan. 2023, [Online]. Available: https://doi.org/10.48550/arXiv.2301.06617

[19] A. Connearu *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," in *Proc. of the 58th Ann. Meeting of the Assoc. for Comp. Linguistics*, 2020 [Online]. Available: https://doi.org/10.48550/arXiv.1911.02116

[20] L. van der Matten and G. Hinton, "Visualizing Data using t-SNE," *Jour. of Mach. Learning Research*, vol. 9, pp. 2579–2605, Nov. 2008.

[21] R. Lim, M. Wu, and L. Miller, "Customizing GPT-3 for Your Application," OpenAI, December 14, 2021 [Online]. Available: https://openai.com/blog/customized-gpt-3/#example

[22] Forefront Team, "GPT-J-6B: An Introduction to the Largest Open Source GPT Model," October 14, 2021 [Online]. Available: https://www.forefront.ai/blog-posts/gpt-j-6b-an-introduction-to-the-largest-open-sourced-gpt-model

[23] M. V. Mendieta, "Predicting Collective Violence from Coordinated Hostile Information Campaigns in Social Media," M.S. thesis, Dept. of Def. Ana., NPS, Monterey, California, USA, 2022 [Online]. Available: https://calhoun.nps.edu/handle/10945/71511

[24] Hugging Face, "mteb/leaderboard." Accessed: May 20, 2023 [Online]. Available: https://huggingface.co/spaces/mteb/leaderboard

[25] K. Song, X. Tan, T. Qin, J. Lu, and T. Liu, "MPNET: Masked and Permuted Pre-training for Language Understanding," in *NeurIPS 2020*, 2020 [Online]. Available: https://www.microsoft.com/en-us/research/publication/mpnet-masked-and-permuted-pre-training-for-language-understanding/

[26] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT Sentence Embedding," *CoRR*, vol. abs/2007.01852, Mar. 2022, [Online]. Available: https://doi.org/10.48550/arXiv.2007.01852

[27] Hugging Face, "sentence-transformers/all-mpnet-base-v2." Accessed: May 20, 2023 [Online]. Available: https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[28] S. R. Tamrakar and C. Silpasuwanchai, "Comparative Evaluation of Transformer-Based Nepali Language Models," PREPRINT (Version 1), Dec. 2022, [Online]. Available: https://doi.org/10.21203/rs.3.rs-2289743/v1

[29] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *CoRR*, vol. abs/1907.11692, Jul. 2019, [Online]. Available: https://doi.org/10.48550/arXiv.1907.11692

[30] L. Tunstall *et al.*, "Efficient Few-Shot Learning Without Prompts," *arXiv*, Sep. 2022, [Online]. Available: https://doi.org/10.48550/arXiv.2209.11055

[31] The GDELT Project, "The GDELT Event Database Data Format Codebook V2.0." February 29, 2015 [Online]. Available: http://data.gdeltproject.org/documentation/GDELT-Event_Codebook-V2.0.pdf

[32] N. Reimers and O. Espejel, "Sentence Transformers," Hugging Face, [Online]. Available: https://huggingface.co/sentence-transformers

[33] Hugging Face, "Tokenizer." Accessed: May 2, 2023 [Online]. Available: https://huggingface.co/docs/transformers/main_classes/tokenizer#transformers.PreTrainedTokenizer

[34] Hugging Face, "Accelerate." Accessed: May 3, 2023 [Online]. Available: https://huggingface.co/docs/accelerate/index

[35] Weights & Biases, “The developer-first MLOps platform.” Accessed: June 2, 2023 [Online]. Available: https://wandb.ai/site

[36] OpenAI, “Tokenizer.” Accessed: May 19, 2023 [Online]. Available: https://platform.openai.com/tokenizer

[37] G. Malato, “Are you still using 0.5 as a threshold?,” Towards Data Science, June 13, 2021 [Online]. Available: https://towardsdatascience.com/are-you-still-using-0-5-as-a-threshold-c5728aa98583

[38] R. Agarwal, “ROC Curves & AUC: The Ultimate Guide,” Built In, August 18, 2022 [Online]. Available: https://builtin.com/data-science/roc-curves-auc

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California