2022-12

PREDICTING COLLECTIVE VIOLENCE FROM COORDINATED HOSTILE INFORMATION CAMPAIGNS IN SOCIAL MEDIA

Mendieta, Milton V.

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

PREDICTING COLLECTIVE VIOLENCE FROM COORDINATED HOSTILE INFORMATION CAMPAIGNS IN SOCIAL MEDIA

by

Milton V. Mendieta

December 2022

Thesis Advisor: Timothy C. Warren
Co-Advisor: Ruriko Yoshida

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** December 2022	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE PREDICTING COLLECTIVE VIOLENCE FROM COORDINATED HOSTILE INFORMATION CAMPAIGNS IN SOCIAL MEDIA	**5. FUNDING NUMBERS**
6. AUTHOR(S) Milton V. Mendieta	
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)

The ability to predict conflicts prior to their occurrence can help deter the outbreak of collective violence and avoid human suffering. Existing approaches use statistical and machine learning models, and even social network analysis techniques; however, they are generally confined to long-range predictions in specific regions and are based on only a few languages. Understanding collective violence from signals in multiple or mixed languages in social media remains understudied. In this work, we construct a multilingual language model (MLLM) that can accept input from any language in social media, a model that is language-agnostic in nature. The purpose of this study is twofold. First, it aims to collect a multilingual violence corpus from archived Twitter data using a proposed set of heuristics that account for spatial-temporal features around past and future violent events. And second, it attempts to compare the performance of traditional machine learning classifiers against deep learning MLLMs for predicting message classes linked to past and future occurrences of violent events. Our findings suggest that MLLMs substantially outperform traditional ML models in predictive accuracy. One major contribution of our work is that military commands now have a tool to evaluate and learn the language of violence across all human languages. Finally, we made the data, code, and models publicly available.

14. SUBJECT TERMS violence prediction, hostile information campaigns, multilingual language models, NLP, social media, deep learning, Twitter.	**15. NUMBER OF PAGES** 101
	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.

PREDICTING COLLECTIVE VIOLENCE FROM COORDINATED HOSTILE INFORMATION CAMPAIGNS IN SOCIAL MEDIA

Milton V. Mendieta
Commander, Ecuadorian Navy
BSSE, United States Naval Academy, 1998
BNS, Universidad Naval Comandante Rafael Moran Valverde, 2007
MSSE, Escuela Superior Politecnica del Litoral, 2015
MFE, Universidad de Guayaquil, 2016
MSS, Universidad de las Fuerzas Armadas (ESPE), 2021

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN DEFENSE ANALYSIS
(IRREGULAR WARFARE)

and

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 2022

Approved by: Timothy C. Warren
Advisor

Ruriko Yoshida
Co-Advisor

Carter Malkasian
Chair, Department of Defense Analysis

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The ability to predict conflicts prior to their occurrence can help deter the outbreak of collective violence and avoid human suffering. Existing approaches use statistical and machine learning models, and even social network analysis techniques; however, they are generally confined to long-range predictions in specific regions and are based on only a few languages. Understanding collective violence from signals in multiple or mixed languages in social media remains understudied. In this work, we construct a multilingual language model (MLLM) that can accept input from any language in social media, a model that is language-agnostic in nature. The purpose of this study is twofold. First, it aims to collect a multilingual violence corpus from archived Twitter data using a proposed set of heuristics that account for spatial-temporal features around past and future violent events. And second, it attempts to compare the performance of traditional machine learning classifiers against deep learning MLLMs for predicting message classes linked to past and future occurrences of violent events. Our findings suggest that MLLMs substantially outperform traditional ML models in predictive accuracy. One major contribution of our work is that military commands now have a tool to evaluate and learn the language of violence across all human languages. Finally, we made the data, code, and models publicly available.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

API	application programming interface
AUC	area under the ROC curve
BERT	bidirectional encoder representations from transformers
BPE	byte pair encoding
CC	Common Crawl corpus
CODA	Coalition for Open-Source Defense Analysis
ConvLSTM	convolutional long short-term memory
CPT	continual pretraining
CT-BERT	COVID-Twitter-BERT
DT	decision trees
EDA	exploratory data analysis
ERGM	exponential family random graph model
FN	false negatives
FP	false positives
FPR	false positive rate
GNN	graph convolutional neural network
GPU	graphics processing unit
GPT	generative pretrained transformer
IRA	Internet Research Agency

IRC	information related capabilities
ISIS	Islamic State
KNN	K-nearest neighbors
LaBSE	Language-agnostic BERT Sentence Embedding
LM	language model
LSTM	long short-term memory
mBERT	multilingual BERT
ML	machine learning
MLLM	multilingual language model
MLM	masked language modeling
MT	machine translation
NLG	natural language generation
NLLB	No Language Left Behind
NLP	natural language processing
NLU	natural language understanding
NPS	Naval Postgraduate School
NSP	next sentence prediction
OIE	operations in the information environment
PTS	pretraining from scratch
RF	random forest
ROC	receiver operating characteristic curve
SNA	social network analysis

SSL	self-supervised learning
SVM	support vector machine
TA	target audience
TLM	translation language modeling
TP	true positives
TPR	true positive rate
TPTLM	transformer-based pretrained language model
TPU	tensor processing unit
UCDP	Uppsala Conflict Data Program
UMAP	Uniform Manifold Approximation and Projection
UMSAB	unified multilingual sentiment analysis benchmark
ViEWS	violence early warning system
XLM-R	cross-lingual model—RoBERTa
XLM-T	cross-lingual model—Twitter
XGBoost	extreme gradient boosting

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

The spread of hostile information is increasingly common on the internet today, and especially in social media, a problem that is becoming a global risk. It has been demonstrated that the "reach of social media penetration" has a positive effect on the outbreak of collective violence (Warren 2015, p. 297), which implies that social media may provide a useful source for signals of impending violence. In this thesis, we construct a model that can accept input from social media messages written in any language and learn to associate forms of discourse with the timing and location of events of collective violence. By examining multilingual language models (MLLM) that operate in a language-agnostic manner, we generate vector embeddings that capture semantic elements more likely to occur in the periods immediately preceding and immediately following violent events. Our aim is to provide a useful representation of violent discourse, which might prove beneficial for predicting violence through social media. The main motivation for using MLLMs is their ability to perform *zero-shot cross-lingual* transfer learning, in which "a model that is fine-tuned on one language can be applied to others without any further training" (Tunstall et al. 2022, p. 87). This ability proves particularly beneficial in social media analysis, which must cover a wide diversity of languages in order to be scaled to global analysis.

The purpose of our work is to compare the performance of three MLLMs—LaBSE, smaller-LaBSE, and XLM-T— that are trained to predict message classes linked to past and future occurrences of violent events. LaBSE is a multilingual embedding model developed by Google which is pretrained in 109 different languages on a corpora extracted from Wikipedia and Common Crawl corpus (CC) (Feng et al. 2020). Smaller-LaBSE is just a smaller version of LaBSE, with fewer parameters, that targets 15 languages (Ukjae 2021). In contrast, XLM-T is an MLLM trained exclusively on Twitter data in 100 languages (Barbieri et al. 2020). All these models are available at the Hugging Face Hub, an open-source repository for the sharing of models developed by the natural language processing (NLP) research community.

The results of our experiments produced four noteworthy findings. The *first* finding confirms our hypothesis that the pre-violence signal in a tweet is stronger near the location of a future violent event. A labelwise receiver operating characteristic (ROC) plot of a Random Forest

(RF) classifier shows that receiver operating characteristic curve (ROC)-area under the ROC curve (AUC) scores decrease as the spatial distance increases, suggesting that the presence of the collective violence signal is stronger near the location of violent events. The *second* finding is that ensemble multilabel classifiers are the best performing traditional machine learning (ML) models across the seven different metrics used in our benchmark. Furthermore, ensemble models trained using a problem transformation approach not only perform better than their problem adaptation counterparts, but they are also computationally faster. Likewise, features extracted from XLM-T yield better performances than LaBSE and smaller-LaBSE. As a result, the best performing traditional ML algorithm is random forest (RF) trained on XLM-T features with a ROC-AUC score of 0.6028.

The *third* finding indicates that when feed-forward classifiers are trained on top of smaller-LaBSE, LaBSE, and XLM-T, they yield similar performances in terms of ROC-AUC scores, each scoring approximately 0.73. In terms of memory footprint, smaller-LaBSE is the best performing model due to a smaller vocabulary size and fewer trainable parameters. Finally, the *fourth* finding, which follows from the previous two, is that deep learning MLLMs outperform traditional ML models by a substantial margin, demonstrating 13% higher out-of-sample predictive accuracy as captured by ROC-AUC scores. This last evidence shows that deep learning MLLMs perform better than traditional ML models for predicting which forms of discourse are most closely associated with collective violence.

Our results suggest that a predictive relationship between social media content and past events of collective violence exists, regardless of the language used in the conversation. We believe that our study could serve as an useful decision-aid tool in the military for predicting violence on a global scale at the operational and strategic levels of war. One major contribution of our work is that military commands now have a tool to evaluate and learn the language of violence across all human languages, thus reducing the negative effects of hostile information campaigns in social media. Finally, we made the code, data, and models publicly available at https://huggingface.co/m2im/, with the hope that this will help the research community advance its efforts in conflict prediction in addition to enabling our warfighters to use the model as a tool to enhance their understanding of the information environment.

References

Barbieri F, Camacho-Collados J, Espinosa-Anke L, Neves L (2020) Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *arXiv* abs/2010.12421, https://doi.org/10.48550/arXiv.2010.12421.

Feng F, Yang Y, Cer D, Arivazhagan N, Wang W (2020) Language-agnostic BERT sentence embedding. *arXiv* https://doi.org/10.48550/arXiv.2007.01852.

Tunstall L, Von Werra L, Wolf T (2022) *Natural Language Processing with Transformers* (O'Reilly Media, Inc., UK).

Ukjae J (2021) Smaller-LaBSE. Github. Accessed October 1, 2022, https://github.com/jeongukjae/smaller-labse.

Warren TC (2015) Explosive connections? mass media, social media, and the geography of collective violence in African states. *Journal of Peace Research* 52(3):297–311.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

First and foremost, I would like to thank my lovely wife, Bertha, and our five children for their unconditional support over the past 18 months. I am grateful to my family for their support throughout my education and career, especially to my parents who always celebrate my accomplishments as if they were theirs. I would also like to thank Professor Timothy C. Warren and Professor Ruriko Yoshida for mentoring me and advising on this research. Their professionalism and academic prowess exposed me to new and relevant methods that will enhance my contribution to the armed services and the research community for the remainder of my career. I am also grateful to the Regional Defense Fellowship Program, the U.S. Navy, and the Ecuadorian Navy for providing me the opportunity to attend this great institution and further my education.

Lastly, I would like to thank my sister, who passed away just before this academic endeavor started. I know she would have been very proud of how everything turned out for me in NPS. This thesis is in memory of her.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

The spread of hostile information via the internet and social media especially is increasingly common and poses a global risk. It has been demonstrated that the "reach of social media penetration" has a positive effect on the production of collective violence (Warren 2015, p. 297). Military commanders face opponents in contested information environments, where making sense of the growing flow of information to generate actionable insights is the key to success. The ability to predict the onset of conflicts is useful to anticipate collective violence and prevent human suffering.

State and non-state actors intentionally exploit the information environment to serve their interests, sometimes crafting information strategies that may lead to the emergence and spreading of violent conflicts. Two pieces of information may convey the same claim but be expressed in different languages and through coordinated forms, such as images, text, tweets, memes, and more. Moreover, actors continuously take advantage of the ever-expanding global network of connectivity to design novel ways to generate and disseminate hostile information. Existing approaches to violence prediction fail to cope with this rapidly evolving landscape.

Models used for policy making need to be explainable, but unfortunately, this explainability sometimes comes at the cost of limiting the models' predictive power. Normally, these models are trained on variables carefully chosen by subject matter experts, a process known as feature engineering, which requires specialized talent and intensive human labor. Furthermore, they are better suited for long-term predictions, and their external validity is limited to the dataset on which they were trained and tested. More specialized models use a social network analysis (SNA) approach. The advantage of network models is that they are amenable to training on social media data, which is the source of information we seek to exploit in our work. However, the nature of interdependency in these networks violates the assumptions of independence of standard statistical models, which makes inference or generalization more difficult (Everton 2012, pp. 229–230).

In recent years, the advent of transformer-based pretrained language model (TPTLM) has

revolutionized the field of natural language processing (NLP) for a variety of tasks. These large pretrained models "combine the power of transformers and self-supervised learning" for encoding complex language information such as syntax and semantic relationships, which is then carried over to downstream tasks (Kalyan et al. 2021, p. 1). Given the initial success in English NLP with monolingual models such as bidirectional encoder representations from transformers (BERT) (Devlin et al. 2018), this work was replicated across other domains and different languages. Nonetheless, these pretrained models tend to exist only for high-resource languages (e.g., English, French, Spanish), which makes this approach less suited for predicting conflict at global scale in a variety of popular languages, tribal languages, and dialects. It would simply make this solution infeasible.

Inspired by the tremendous success of monolingual models, the NLP research community developed a class of multilingual language model (MLLM)s, which are pretrained similarly as their monolingual counterparts, except that the corpora contain documents across many languages. Multilingual NLP systems are designed to handle user input in more than one human language, which is beneficial in situations where the system will be used by speakers of different languages. Predicting worldwide collective violence seems to follow this intuition. The surprising performance of MLLMs is achieved by enabling *zero-shot cross-lingual transfer learning*, in which "a model that is fine-tuned on one language can be applied to others without any further training" (Tunstall et al. 2022, p. 87). The authors further claimed that in some cases "this ability to perform cross-lingual transfer produce results that are competitive with those of monolingual models, which circumvents the need to train one model per language" (Tunstall et al. 2022, p. 92).

Despite these remarkable cross-lingual features, most MLLMs are pretrained on formal text, which make them ill-prepared to capture the informal discourse of social media (i.e., slang, emojis, misspellings, vulgarisms). Social media, such as Twitter, provide an enormous source of information worldwide that could be used in violence prediction models. In fact, we hypothesize that there exists a sufficient collective violence signal in the online discourse prior to the occurrence of violent conflicts. We further argue that the closer the conversation is to these conflicts, in terms of time and distance, the stronger the presence of this signal will be. This hypothesis drives our modeling process and the corresponding prediction outcomes. We test this hypothesis by first constructing our own multilingual corpus based on historical Twitter data and a spatial/temporal heuristic around past violent conflicts, and then building

a multilingual classifier using three MLLMs. The first model, called Language-agnostic BERT Sentence Embedding (LaBSE), is pretrained on regular text in 109 languages, while the second model, called cross-lingual model—Twitter (XLM-T), is fine-tuned specifically on Twitter data in 100 languages. The last model, dubbed the smaller-LaBSE, is a smaller version of LaBSE that targets 15 languages. If a predictive relationship is found between social media content and past events of collective violence, our study could serve as a useful decision-aid tool for predicting violence on a global scale.

With more than 7,000 languages spoken in the world today, our solution calls for a model that is language-agnostic in nature (David et al. 2022). This approach will considerably reduce the complexity associated in dealing with language-specific models. Our work builds on an existing project in the Coalition for Open-Source Defense Analysis (CODA) Lab at Naval Postgraduate School (NPS) for the automated detection of hostile information campaigns using artificial neural networks (Warren and Barreto 2020). The current approach considers a *U-Net* deep neural network that predicts violence on a global scale, using seven different inputs, one of which is a raw count of georeferenced Twitter messages. By dividing the surface of the Earth into approximately 20 million grid cells (approx. 5 km x 5 km), the model is trained to predict the occurrence of collective violence in each cell on a week-to-week basis (Warren and Barreto 2020). Our goal is to develop metrics which can contribute to this effort by improving the predictive power of the Twitter data inputs that are fed into the network by leveraging MLLMs.

Thus, our study focuses on answering the following research question, "To what extent can the information contained in social media (Twitter) generate sufficient signals for use in multilingual transformer models to predict collective violence events efficiently?" We postulated this question because there is a gap in understanding signals of collective violence in multiple or mixed languages in social media. A desired solution for predicting collective violence would be an approach that is used for short-term prediction, language-agnostic, and trained using social media data without requiring labor-intensive manual annotations. This approach is what this project intends to address. The purpose of this study is twofold. First, it aims to collect a multilingual violence corpus from archived Twitter data using a proposed set of heuristics that account for spatial-temporal features around past and future violent events. And second, it attempts to compare the performance of traditional machine learning (ML) classifiers against deep learning MLLMs for predicting message classes

linked to past and future occurrences of violent events. Our findings suggest that MLLMs outperform traditional ML models by approximately 13% in terms of ROC_AUC scores. One major implication of our work is that military commands now have a tool to evaluate and learn the language of violence across all human languages, thus reducing the negative effects of hostile information campaigns in social media.

This thesis contains five chapters: the first chapter has introduced the problem and highlights the need for predicting collective violence events. The second crafts a comprehensive literature review concerning different conflict prediction techniques and current state-of-the-art multilingual models in the NLP community. The third devises the methodology for building our multilingual corpus, as well as the process of pretraining and fine-tuning the MLLMs used in our experiments. Finally, we analyze different multilingual settings in diverse conflict environments to determine the model best suited for the task of predicting collective violent events. The conclusions explore the implications of these findings and suggest further avenues of approach to tackle this problem more effectively. The code, data, and models are made publicly available at https://huggingface.co/m2im/, with the hope that this will help the research community advance its efforts in conflict prediction in addition to enabling our warfighters to use the model as a tool to enhance their understanding of the information environment.

CHAPTER 2:
Literature Review

Violence prediction from coordinated hostile information campaigns in social media is a topic that is getting a lot of traction in the the research community. Methods abound in the literature that take on a data-driven approach, including statistical and machine learning models, deep learning, and social network analysis. We claim that the use of MLLMs represents a turning point for the task of conflict prediction. For what follows in the remainder of this chapter, we organized the literature review into three different sub-sections based on relevant topics: coordinated hostile information campaigns, violence prediction, and multilingual language models.

2.1 Coordinated Hostile Information Campaigns

Some may erroneously assume that a hostile information campaign conveys messages solely based on false statements. However, in reality, a hostile information campaign may also consist of truthful statements, or statements that the sender believes to be true, though the ultimate goal is to incite collective violence. For example, when a rebel group questions the legitimacy of the state, or points to ways in which certain minorities have been oppressed, or says that another ethnic group poses a threat, these are all statements designed to promote collective violence, but they are not necessarily *false statements*, even though state representatives may label them as such. Similarly, when the Russian Internet Research Agency (IRA)[1] promoted messages created by activists in the Black Lives Matter movement, these were messages that the original senders believed to be true, even though they were also messages that Russian operatives thought could be leveraged to increase division and hostility. Hence, the first step in this research effort is to define the subject of inquiry, hostile information campaigns.

The term hostile information campaign overlaps with many other related concepts, such as misinformation, disinformation, rumors, fake news, propaganda, and others. The public in

[1]IRA is a Russian organization engaged in online propaganda and influence operations on behalf of Russian political leaders.

general, and researchers in the field in particular, tend to use these terms interchangeably because the boundaries are somewhat blurry. Let us start with the first related definition, misinformation. According to Islam et al., misinformation is defined as "a false statement to lead people astray by hiding the correct facts" (Islam et al. 2020, p. 81). Wu et al. go one step further by defining four types of misinformation:

> - *Rumor* refers to unverified information that can be either true or false.
> - *Fake news* refers to false information in the form of news (which is not necessarily disinformation since it may be unintentionally shared by innocent users).
> - *Spam* refers to irrelevant information that is sent to a large number of users.
> - *Disinformation* also refers to inaccurate information which is usually distinguished from misinformation by the intention of deception. (Wu et al. 2019, p. 1)

According to the previous definitions, a hostile information campaign is a mixture of disinformation, rumors, and false information whose purpose is to incite violence. A broader term that stresses the role of information as the essential motivator for the target audience to undertake aggressive and hostile actions is the one proposed by Mazarr et al. (2019) called hostile social manipulation. The authors defined hostile social manipulation as "the purposeful, systematic generation and dissemination of information to produce harmful social, political, and economic outcomes in a target area by affecting beliefs, attitudes, and behavior" (Mazarr et al. 2019, p. 15). One distinguishing factor in this definition is that hostile social manipulation targets the cognitive domain as opposed to military objectives, which is the intended outcome we are trying to explore in this project.

The last definition is very close in spirit to the concept of operations in the information environment (OIE) in the United States Military Joint doctrine: "the application, integration, and synchronization of information related capabilities (IRC)s to influence, disrupt, corrupt, or usurp the decision making of TAs [target audiences] to create a desired effect to support achievement of an objective" (Joint Chiefs of Staff 2006); *Information Warfare* in Russian doctrine: "a holistic concept that includes computer network operations, electronic warfare,

psychological operations and information operations" (Popescu and Secrieru 2018, p. 67); and the Chinese *Unrestricted Warfare* doctrine: "warfare that transcends all boundaries and limits. . . going from weapons systems symbolized by gunpowder to those symbolized by information" (Qiao et al. 2002, p. 12). Some of these concepts are referenced in the literature under different names, such as hybrid warfare, gray zone activities, and active measures. All of these definitions can be summarized as hostile non-kinetic actions in the cognitive domain aimed at the perceptions of the adversaries, two key ideas that must be incorporated in any formal definition of hostile information campaigns.

Campaigns of social manipulation and hostile information share similar goals, objectives, and techniques. Mazarr et al. proposed nine different goals and objectives for social manipulation campaigns, from which only one relates to our current project: "Generate conflict and tension among components of target society" (Mazarr et al. 2019, p. 19). Moreover, social manipulation campaigns in general, and hostile information campaigns in particular, frequently "tap into well-established belief systems and social grievances for its effects," making use of a wide variety of tools for influencing the intended target (Mazarr et al. 2019, p. 8). Some of those tools pertaining to social media are computational propaganda (bots), social media commenting, trolling, astroturfing, and social influencer campaigns (Mazarr et al. 2019, pp. 22–23). One of the key findings in a recent study about radio and cellular communications infrastructures in 24 African states also showed a direct connection between social media and collective violence: "the reach of social media penetration generates substantial increases in collective violence, especially in areas lacking access to mass media infrastructure" (Warren 2015, p. 297). It follows from the previous statements that social media information is a key feature for the success of any predictive model of violence.

For the purpose of this project, we propose the following definition of hostile information campaigns, which is adapted from the previous definition of hostile social manipulation by Mazzar et al. as follows: *Hostile information campaigns* are the "purposeful, systematic generation and [coordinated] dissemination of information to produce . . . [social unrest] in a target [population] by affecting beliefs, attitudes and behavior [that lead to collective violence]"(2019, p. 15). The main difference in this definition is the use of *information* rather than *misinformation* to encompass both true and false statements that could be equally likely to incite collective violence.

2.2 Violence Prediction

There have been many approaches in the literature that have exploited the power of statistical and machine learning models to predict the outcome of violence. For example, in a study for the World Bank, Celiku and Kraay evaluated the predictive power of two unconventional binary linear classification algorithms "in a set of conflict and non-conflict episodes constructed from a large country-year panel of 144 developing countries since 1977" (2017, p. 2). Their approach was to minimize the losses from both classifiers, which is "the same prediction loss function that is also used to evaluate the quality of the predictions" (Celiku and Kraay 2017, p. 3). Both models were trained on many features in three broad categories: latent tensions (e.g., satellite night light density, GDP), shocks (e.g., changes in terms of trade, number of neighboring countries in conflict), and institutions (e.g., political terror scale, Freedom House composite indicator of civil liberties and political rights) (Celiku and Kraay 2017, pp. 20–21). The best classifier—dubbed the threshold classifier—achieved a modest performance, and the authors claimed that one of the main limitations of their approach was the lack of generalization in datasets different from the ones used for training and testing.

Similarly, Hegre et al. proposed a dynamic multinomial logistic regression model that "predicts changes in global and regional incidences of armed conflict for the 2010 — 2050 period" (2013, p. 250). The model was trained "on a 1970 -– 2009 cross-sectional dataset of changes between no armed conflict, minor conflict, and major conflict" (Hegre et al. 2013, p. 250). Exogenous variables (i.e., population size, infant mortality rates, demographic decomposition, education levels), as well as endogenous variables (i.e., conflict, recent conflict history, and neighboring conflicts) fed the model during training. The authors estimated a 0.937 area under the ROC curve (AUC) score on an out-of-sample test set for positive prediction, and claimed "a continued decline in the proportion of the world's countries that have internal armed conflict from about 15% in 2009 to 7% in 2050" (Hegre et al. 2013, p. 250). Despite these models achieving acceptable performances, they are better suited for long-term predictions.

A number of studies have gone beyond these early long-term prediction efforts to consider conflict prediction at smaller spatial resolutions and shorter timescales. Two theses at NPS use sentiment in Twitter messages to predict collective violence using regression models.

Similar to our work, both studies are limited to the period from August 1, 2013, through July 31, 2014, to match the timespan of a historical archive of Twitter messages licensed for research at NPS. Frost et al. examined five independent variables (road distance, night light emissions, population density, previous violent events, and deaths) and three dependent variables (negative Twitter messages, extremely negative Twitter messages, and total Twitter messages) in four different negative binomial regression models whose predictions were constrained to "2km grid cell-months" (2017, p. 27). The study was restricted to Iraq and showed that "extremely negative terminology seems to be more useful than moderately negative terminology in generating accurate predictions of violent events" (2017, p. 39). Likewise, Kuah et al. proposed six non-linear Poisson regression models for sentiment analysis related to the Euromaidan movement in Ukraine, whose unit of analysis was set to "a grid-cell width of approximately 20 kilometers" (2018, p. 24). Similar to Frost et al., this study also showed a significant correlation between Twitter metrics and the outcomes of violent events (2018, p. 40).

Similarly, Hegre et al. present a model dubbed the violence early warning system (ViEWS) for predicting political violence (Hegre et al. 2019). ViEWS is restricted to Africa and generates predictions 36 months into the future at two levels of analysis: "country-months (*cm*) and subnational geographical location months (*pgm*) . . . that cover all areas of the world at a resolution of 0.5 x 0.5 decimal degrees" (Hegre et al. 2019, p. 157). The model is an ensemble model of other available models in the literature. For example, there are 24 models in the *cm* ensembles and 30 in *pgm*. In a follow-on work, Hegre et al. introduced two important updates to ViEWS: "(1) a new infrastructure for training, evaluating, and weighting models . . . , and (2) a number of new forecasting models that contribute to improve overall performance, in particular with respect to effectively classifying high- and low-risk cases." (2021, p. 599). These enhancements provided a benchmark for adding or retaining models in the final ensemble. In both cases, the authors made the data and source code publicly available.

Python et al. proposed a theoretically informed model that produces short-term predictions of non-state terrorism worldwide (2021). The authors divided the world into 13 regions and built individual models for each region—generalized additive model (GAM), extreme gradient boosting (XGB) and random forest (RF)— under the claim that "training machine learning models for separate regions allows the algorithms to select different hyperparame-

ters for different regions (e.g., regions with lower prevalence require greater regularization)" (2021, p. 1). To a lesser geographic scope, Brandt et al. focused on monthly-grid predictions for Africa using a model that combines graph convolutional neural network (GNN) for capturing spatial features and long short-term memory (LSTM) for capturing temporal dependencies (2022). Similarly, Radford et al. followed the same approach but used a convolutional long short-term memory (ConvLSTM) model to predict the outcomes of violence in Africa using the ViEWS benchmark.

While few studies have focused on uncovering coordinated hostile information campaigns that signal an increase in collective violence, the literature abounds for topics related to the various types of misinformation, which can be easily extrapolated to our subject of inquiry. Xu et al. argue that detecting misinformation is a challenging task because the model needs to capture the relationship between the reported information and the real information (Xu et al. 2019). Existing approaches use SNA to uncover coordinated networks in social media. For example, Pacheco et al. introduced a general, unsupervised network-based methodology using bipartite networks "to uncover groups of accounts that are likely coordinated" (2021, p. 1). The authors used Twitter accounts to examine "identities, images, hashtag sequences, retweets and temporal patterns" (Pacheco et al. 2021, p. 1). They presented few influence campaign studies of different backgrounds (i.e., U.S. elections, Hong Kong protests, Syrian Civil War, and cryptocurrency manipulation). Note that this approach relies on the use of metadata, and not the actual Twitter text content, except for images.

Similarly, Weber and Neuman also proposed a method that relies on metadata alone. The authors built "an undirected weighted network of accounts, which is then mined for community extraction" (2021, p. 1). In contrast, Vargas et al. took on a different twist by proposing a model that uses "a time series of daily coordination networks for distinguishing the activities of a disinformation campaign from legitimate Twitter activity" (2020, p. 1). The authors used these neworks as feature extractors and then trained a binary classifier on top. Once again, this model is trained on Twitter metadata only.

Most SNA approaches based on centrality metrics focus on identifying accounts that spread misinformation; however, there are statistical frameworks such as exponential family random graph model (ERGM) that seek to exploit the network structural features by predicting tie formation. For example, Williams et al. (2020) applied ERGMs to understand the dis-

semination of political propaganda using Twitter bots during the riots in Ecuador in October 2019. The authors gathered tweets using the hashtag *#YoTambienSoyZangano* during October 5, 2019, then tested for homophily and transitivity between bots and humans, suggesting that bots are less likely to form links with humans than with other bots. The main limitation of these approaches is that they do not rely on the actual Twitter content, and more so ERGM whose statistical assumptions make generalization more difficult (Everton 2012, pp. 229–230).

However, a number of recent studies have used content-based metrics from social media for violence prediction. Some authors used Twitter data from the 2015 Baltimore protests to discuss how social networks can inflame violent protests (Mooijman et al. 2018). The authors manually annotated 4,800 tweets as having moral content "to operationalize online moral rethoric" to train some classification models further down the pipeline (2018, p. 390). The study shows that the variance in moral endorsement towards a protest changes over time, and hence these endorsements can be measured to predict the outcome of violence.

Alizadeh et al. took on a different approach by training an RF classifier for distinguishing influence campaign content from normal activity in social media (Alizadeh et al. 2020). The model took as inputs some human-interpretable features (i.e., word count, topic of a post, sentiment) obtained from message content. Under the claim that "Industrialized production of influence campaign content leaves a distinctive signal in user-generated content that allows tracking of campaigns from month to month and across different accounts" (2020, p. 1), the authors ran five different tasks with the same model over multiple short time intervals to assess how the influence campaigns behave over time.

Likewise, Muller et al. studied the association "between antirefugee sentiment on Facebook and hate crimes against refugees in Germany" using posts from the Facebook page of the *Alternative für Deutschland* anti-refugee and anti-inmigration political party (Müller and Schwarz 2021, p. 2132). And finally, Mitts et al. (2022) analyzed the effect of Islamic State (ISIS) propaganda on social media using a ML approach. The authors examined the Twitter discourse between 2015 and 2016 before most of the accounts related to ISIS were banned from social media. Since most of the propaganda included audiovisual content, the proposed recruitment detection model relied first on audio transcriptions before applying text classification techniques. The study shows that "propaganda conveying the material,

spiritual, and social benefits of joining ISIS increased online support for the group, while content displaying brutal violence decreased endorsement of ISIS" (Mitts et al. 2022, p. 1).

2.3 Multilingual Language Models

The availability of vast amounts of unannotated data, the evolution of better graphics processing unit (GPU), and the development of the transformer architecture gave rise to the increased use of deep learning models for NLP systems, more specifically TPTLM. At a high level, transformer models allow us to work with sequence data. They are composed of an embedding layer, encoder-decoder layers, and the attention mechanisms proposed by Vaswani et al. (2017). To reveal the complexities associated with TPTLMs in general, and MLLMs in particular, for what follows in this section, we propose analyzing these models under the following taxonomy: architecture, transfer learning, pretraining tasks, pretraining methods, fine-tuning, model capacity, vocabulary, and pretraining corpus.

2.3.1 Architecture

Although there are numerous transformer models in the literature, most of them belong to three different architectures. *Encoder-only* models such as BERT (Devlin et al. 2018) and its variants, like RoBERTa (Liu et al. 2019) and ALBERT (Lan et al. 2019), and other BERT-based models like LaBSE (Feng et al. 2020) belong to this category. These models are well suited for natural language understanding (NLU) tasks such as text classification, known as named entity recognition and sentiment analysis. *Decoder-only* models are *generative* in nature and best suited for natural language generation (NLG) tasks; given a prompt of text they will try to predict the next token. The family of generative pretrained transformer (GPT) models like GPT-1 (Radford et al. 2018), GPT-3 (Brown et al. 2020) belong to this category. *Encoder-decoder* models are used for mapping one sequence of text to another; they are well suited for machine translation and summarization tasks. BART (Lewis et al. 2019), T5 (Raffel et al. 2020) and their multilingual counterparts mBART (Liu et al. 2020a) and mT5 (Xue et al. 2020) belong to this class. For what follows, our analysis is limited to encoder models only, since those architectures are the best suited for our task of predicting violence (e.g., text classification). Any mention of a model with a different architecture will be solely for the purposes of comparing, contrasting, or illustrating a concept.

2.3.2 Transfer Learning

Transformer models, including monolingual language models and MLLMs, rely on transfer learning to reuse the knowledge learned in a source task as the starting point for a model on a second task. Transfer learning is enabled via self-supervised learning (SSL), a new learning paradigm where models identify hidden patterns and learn universal signals from massive unlabeled data to generate labels automatically based on pretraining tasks. Extending this concept from monolingual models, MLLMs learn embeddings with high overlap across languages through common shared multilingual representations (Doddapaneni et al. 2021, p.15).

As a result, a myriad of MLLMs have been proposed by the NLP community stemming from their monolingual state-of-the-art counterparts. For example, multilingual BERT (mBERT) from BERT (Devlin et al. 2018), XLM-R (Conneau et al. 2019) from RoBERTa (Liu et al. 2019), mBART (Liu et al. 2020a) from BART (Lewis et al. 2019), just to name a few. This surprising ability to generalize well across different languages is what makes MLLMs good candidates for tackling our worldwide conflict prediction problem. This nice generalization feature cannot be confused with the misconception in the NLP community that MLLMs learn universal language patterns. Doddapaneni et al. (2021, p. 2) conducted a series of experiments to test this hypothesis, concluding that there is no consensus yet to support this claim.

2.3.3 Pretraining Tasks

TPTLMs yield better performances when the pretraining task, or training objective function, resembles the objective of the corresponding downstream task, thus closing the learning gap between pretraining and fine-tuning (Kalyan et al. 2021). A wide variety of objective functions abound in the literature for training MLLMs. For example, masked language modeling (MLM) is the most standard type of SSL that seeks to predict masked tokens based on unmasked tokens encoded with bidirectional context (Kalyan et al. 2021, p. 4). Models such as BERT and mBERT (Devlin et al. 2018), RoBERTa (Liu et al. 2019), and LaBSE (Feng et al. 2020) are pretrained with MLM. An extension of MLM for a multilingual setting that uses monolingual as well as parallel data is translation language modeling (TLM), introduced by (Lample and Conneau 2019). Similar to MLM, here in TLM the masked tokens incorporate context from the parallel corpora. LaBSE (Feng et al.

2020) is pretrained using TLM.

The next sentence prediction (NSP) pretraining task is a binary sentence-pair classification task used by BERT, which allows the model to understand longer-term dependencies across sentences by identifying whether a given pair is consecutive or not (Devlin et al. 2018). MLM, TLM, and NSP are suited for text-classification related tasks, such as topic classification, language identification, and sentiment analysis; however, Barbieri et al. argue that NSP is not appropriate for Twitter data where most tweets are composed of a single sentence (2020, p. 3). Furthermore, Wang et al. (2019) studied the effect of the NSP learning objective for the cross-lingual ability of mBERT, concluding that this pretraining task hurts the model performance. It follows from this evidence that we should avoid using NSP during any pretraining process when building our violence prediction classifier.

2.3.4 Pretraining Methods

Pretraining transformer models is computationally prohibitive and requires a huge amount of unlabeled data. Kalyan et al. (2021, pp. 7–9) suggested five types of pretraining methods for TPTLM, but for the sake of this project, I will refer only to the two most widely used approaches: pretraining from scratch (PTS) and continual pretraining (CPT). PTS involves training the models by randomly initializing the model parameters and then learning these weights by minimizing the losses of the corresponding pretraining tasks (e.g., MLM, TLM, NSP). BERT (Devlin et al. 2018), RoBERTa (Liu et al. 2019), and SciBERT (Beltagy et al. 2019) belong to this training paradigm; the latter model follows the same architecture as BERT, except that it is trained on scientific data instead. PTS requires a large number of GPUs or tensor processing unit (TPU) and long compute times. For example, Barbieri et al. (2020) proposed a MLLM dubbed XLM-T, a RoBERTa model pretrained on 60M tweets which converged after eight to nine days on eight NVIDIA V100 GPUs. The authors claimed that the estimated cost for training this language model (LM) was USD 4,000 on Google Cloud.

CPT, in contrast to PTS, starts the training from existing language model parameters that are updated after further pretraining to the target domain. Generally speaking, CPT is commonly used to develop domain-specific models like BioBERT (Lee et al. 2020), which is initialized from general BERT and further pretrained on a biomedical corpus. The

multilingual landscape has also benefited from CPT. For instance, PortugueseBERT (Souza et al. 2020) is initialized from the internal states of mBERT and further pretrained with its own monolingual corpus.

CPT requires fewer computing resources and training time than PTS. For example, SciBERT was trained from scratch on the SCIVOCAB for one week on a single TPU v3 with eight cores (Beltagy et al. 2019), and BioBERT was trained with CPT for over 20 days with eight V100 GPUs (Lee et al. 2020). For the sake of a fair comparison, since both models were trained in different infrastructures, we can estimate their relative performance using the following approximation: training BERT large on 64 GPUs (e.g., V100s/RTX 2080 Tis)—the equivalent of 16 TPUs—takes approximately five days.[2] All things being equal, the choice of using one pretraining method over the other, meaning PTS or CPT, is mainly limited to the amount of computing resources available. This last statement assumes that we have access to a large volume of unlabeled data: "a few hundred MiB of text data is usually a minimal size for learning a BERT model" (Conneau et al. 2019, p. 3).

2.3.5 Fine-Tuning

In many practical data-constrained applications we do not have access to enough compute resources, large amounts of text data, or the amount of labeled training data is limited. How do we cope with these constraints? The answer is *fine-tuning*. There seems to be some confusion about the terms *pretraining* and *fine-tuning* in the NLP community; they are frequently used interchangeably, but despite the similarities, in practice they are different. Technically speaking, both methods are similar because they perform some sort of transfer learning; this results in weight updates where the model parameters are changed. However, they are different because *pretraining* refers to an SSL paradigm that uses massive *unlabeled* data to gain common background knowledge, as opposed to *fine-tuning* that relies on relatively small *labeled* datasets used for downstream tasks that require target-specific knowledge.

One must be aware that fine-tuning using a small dataset on a large pretrained model (a model with lots of parameters) is either prone to overfitting, or the model will perform

[2]For more details about these calculations, please refer to this excellent blog available at https://timdettmers.com/2018/10/17/tpus-vs-gpus-for-transformers-bert/.

poorly if the weights are not adapted well to the task at hand (Kalyan et al. 2021, p. 25). Liu et al. (2020b) suggest using a continual learning approach when fine-tuning a model in order to preserve the cross-lingual ability learnt during pretraining. Simply put, fine-tuning benefits from CPT. Given that we possess a large unlabeled corpus that can be converted into a labeled dataset, then any potential violence classifier should devise a fine-tuning strategy enhanced by CPT to the greatest extent possible.

2.3.6 Model Capacity

Somewhat surprisingly, the performance of TPTLM appears to obey a set of scaling laws proposed by Kaplan et al. (2020), who state that a more productive path towards better models is to focus on increasing model capacity (number of parameters), pretraining compute time, and pretraining training data, all three in tandem to achieve optimal performance. This observation triggered the development of bigger models and bigger datasets. Evidence in the field shows that in the almost five years since the release of the original transformer architecture in 2017, the size of transformer models has increased by over four orders of magnitude. Just to have an idea of this scaling explosion, let us revisit the following examples:

1. BERT was trained on English text data (16GB) using 2,500M words from Wikipedia and 800M words from *Book corpus*, the base and large versions have 110M and 335M parameters, respectively.
2. The cross-lingual model—RoBERTa (XLM-R) (Conneau et al. 2019) is the result of the following work of XLM (Lample and Conneau 2019) and RoBERTa (Liu et al. 2019), taking pretraining one step further by massively scaling to 2.5 TB of training data gathered from the Common Crawl corpus (CC), yielding a model with 270M and 550M parameters for the base and large versions, respectively.
3. GPT-3, a descendant of the GPT family, is a scaled-up version of the original models without many architectural modifications; it was trained on hundreds of billions of words coming from different data sources (e.g., Wikipedia, CC, Books, Webtext), resulting in a model with 175 billion parameters whose checkpoints require 800GB of storage capacity.
4. Two giant experimental models developed by Google called *Gshard* (Lepikhin et al. 2020) and *Switch Transformers* (Fedus et al. 2022) claim an impressive amount of

600B and 1.6T parameters each.

2.3.7 Pretraining Corpus

More than 7,000 languages are spoken in the world today, but machine translation (MT) systems have been built for approximately 100 languages (David et al. 2022). MLLMs are trained either on monolingual or parallel data: for each sentence in a language, we have the same sentence in a different language. Some MLLMs support a large number of languages. For example, the first multilingual model mBERT was released by the authors of BERT a month after they published and shared the original English BERT model. The authors simply added a *readme* section in the Github repository where they explained what they did for pretraining mBERT on Wikipedia in 104 different languages. XLM-R is trained on CC-100—a collection of 2.5 TB of text data for 100 languages—using more text data for low-resource languages. Two larger models based on XLM-R were developed later—dubbed XLM-RXL and XLM-RXXL—also trained on CC-100 with improved performance gains (Goyal et al. 2021). MLLMs also support a smaller set of languages: Unicoder (Huang et al. 2019) was trained on monolingual (Wikipedia) and parallel data for 15 languages, MuRIL (Khanuja et al. 2021) was trained on CC + Wikipedia targeting 17 Indian languages, and IndoBART (Cahyawijaya et al. 2021) designed for three widely spoken languages of Indonesia: Indonesian, Javanese, and Sundanese.

Despite the fact that MLLMs are known to suffer from the *curse of multilinguality*, "more languages leads to better cross-lingual performance on low-resource languages up until a point, after which the overall performance on monolingual and cross-lingual benchmarks degrades" (Conneau et al. 2019, p. 1), Facebook and Google have taken the multilingual landscape to its limits. Facebook open sourced the code, datasets, and training procedure for a model called No Language Left Behind (NLLB), very similar to GPT-3 but with some interesting architectural modifications to accommodate 200 languages. The model was trained on a 200 paired-language dataset. Google, on the other hand, published a paper claiming the development of a model that supports 1,629 languages, yet the model has not been released (Siddhant et al. 2022). The main contribution of this paper was the construction of dataset itself, along with the engineering challenges to get a model this big to work.

A model developed by researchers at Google AI called LaBSE deserves further exploration. LaBSE is a multilingual embedding model that produces dense vector representations from 109 different languages into a shared embedding space, and it is pretrained using MLM and TLM objectives (Feng et al. 2020). Training on 17 billion monolingual sentences and six billion bilingual sentence pairs results in a model that performs particularly well on low-resource languages not seen during training (Feng et al. 2020). There are interesting applications of LaBSE with promising results. Lin et al. (2021) proposed a multilingual text classification of Dravidian languages (26 South Indian languages), trained on two different datasets for tasks related to polarity classification and offensive language identification. Pei et al. (2022) used LaBSE to encode the input data for the tasks of sentiment analysis and emotion analysis from the low-resource Uyghur language. Finally, a a case more relevant to our research project is the work done by Gencoglu (2020), in which the author performed a large-scale, language-agnostic discourse classification of tweets during COVID-19 . The model was trained for discourse classification on 26 million COVID-19 tweets in different languages, collected during the early stages of the global spread from January 4, 2020, to April 5, 2020. LaBSE was used to encode the training data, ending up with vectors of length 768 for each observation, which were then passed to a machine learning classifier to classify text into semantic categories.

Nevertheless, the promises of LaBSE as a good method to get sentence embeddings across 109 different languages come at a high cost. The model is hard to fine tune due to its 471 million parameters. Fortunately, Ukjae (2021) proposed a model called smaller-LaBSE —a smaller version of LaBSE— whose parameter size and vocabulary are approximately 45% and 35% of the original model, respectively. In contrast to any of the transformer models discussed thus far, smaller-LaBSE is neither pretrained nor fine-tuned on any dataset. Instead, the authors obtained the model by following closely a procedure proposed in a seminal paper titled *Load What You Need: Smaller Versions of Multilingual BERT*, which is based on the fact that for MLLMs, "most of the parameters are located in the embedded layer. Therefore, reducing the vocabulary size should have an important impact on the total number of parameters" (Abdaoui et al. 2020, p. 1). In the case of LaBSE, the word embedding table is approximately 385 million parameters, approximately 81% of the total number of parameters in the model (Ukjae 2021). The author claimed that after evaluating smaller-LaBSE for 14 languages on the Tatoeba dataset, the performance drop when compared to the

original model was not significant (Ukjae 2021). Moreover, smaller-LaBSE targets 15 high-resource languages that are widely spoken in the world: English, French, Spanish, German, Chinese, Arabic, Italian, Japanese, Korean, Dutch, Polish, Portuguese, Thai, Turkish, and Russian (Ukjae 2021).

2.3.8 Vocabulary

Typically, MLLMs which support more languages have a larger vocabulary. A vocabulary is built by the tokenizer specific to the model; it is the union of all tokens identified across all languages. The following are examples of some sub-word tokenization methods used in NLP: Byte Pair Encoding (BPE) (Sennrich et al. 2015), Byte Level BPE (bBPE) (Radford et al. 2019), SentencePiece (Kudo and Richardson 2018), and WordPiece (Wu et al. 2016). Among these tokenizers, only SentencePiece tokenizes the raw text directly, and it does so by not using space as a separator, which is more suited in multilingual settings where not all languages are space segmented. BERT, mBERT, and LaBSE use Wordpiece as tokenizers, whereas XLM-R uses SentencePiece. In the world of transformers, sub-word or character tokenization is preferred over word tokenization, because the former can more efficiently handle unknown words and yield smaller vocabulary sizes.

Despite the myriad of successful multilingual applications today, the social media landscape—Twitter specifically—seems to have been neglected by this trend. In terms of the pretraining corpus, existing models trained on formal text do not always perform well on social media applications (Kalyan et al. 2021). Nonetheless, there have been few attempts to develop Twitter-based TPTLM. One of the most prominent works is the one presented by Barbieri et al. (2020), where the authors proposed a unified benchmark called *tweeteval*, consisting of seven Twitter-specific classification tasks in English and then evaluated it under three different strategies: 1) a pretrained RoBERTa model as is, 2) training RoBERTa from scratch—dubbed RoBERTa-Twitter—using 60 million of English Twitter data, and 3) fine-tuning RoBERTa—dubbed RoBERTa-RT—using the same Twitter data. Curiously, the RoBERTa-Twitter model did not perform as well as the fine-tuned RoBERTa-RT model, perhaps due to the small sample of 60 million tweets used during pretraining: "using a pre-trained LM may be sufficient, but can improve if topped with extra-training on in-domain data" (Barbieri et al. 2020, p. 5).

Bertweet (Nguyen et al. 2020) was trained from scratch on 850 million English tweets using the same BERT-base architecture but leveraging the RoBERTa training procedure. The authors then fine-tuned this model on 23 million COVID tweets, yielding two variants: Bertweet-COVID-cased and Bertweet-COVID-uncased. All models achieved state-of-the-art performances in various tasks, including text classification. COVID-Twitter-BERT (CT-BERT) is another model based on English BERT-large, which was then fine-tuned on a corpus of 160 million tweets about coronavirus (Müller et al. 2020). The model was evaluated on five different classification tasks, yielding better performances than BERT-large, especially in health-related content, as would be expected. Barbieri et al. (2022) proposed XLM-T, a multilingual model based on XLM-R that was pretrained using 198 million tweets in 30 different languages. The authors also proposed a set of unified Twitter datasets in eight different languages called unified multilingual sentiment analysis Benchmark (UMSAB), where XLM-T was further fine-tuned on this benchmark for sentiment analysis tasks, resulting in a model dubbed XLM-T-Sent.

In a follow-on work, Antypas et al. (2022) used XLM-T-Sent in a large-scale multilingual political sentiment analysis project. The authors fine-tuned the model even further on their own manually annotated dataset with tweets from politicians in Greece, Spain (Catalonia, Basque country), and the United Kingdom (Northern Ireland, Scotland, Wales). Their results suggested that negatively charged tweets spread more widely and are more correlated with popularity. Their work is similar to this research effort in the sense that it uses a multilingual approach; however, it is limited to sentiment analysis tasks in a few languages, as opposed to a multilabel classification setting supporting all languages observed on Twitter. Apart from Bertweet, all other models were pretrained using CPT, and the only Twitter-MLLM is XLM-T, whose pretrained version supports over 30 languages.

Table 2.1 summarizes the most important features for the monolingual and multilingual language models analyzed in this section. For a more thorough survey of TPTLM models in general see (Kalyan et al. 2021), and for MLLMs in particular see (Doddapaneni et al. 2021). A careful examination of Table 2.1 reveals that models trained on Twitter data are mostly monolingual. To the best of our knowledge, XLM-T is the only MLLM trained on Twitter data for text classification; however, its training corpus is far from violence-related discourse. The use of XLM-T—or in fact any other MLLM like LaBSE—for predicting violence will therefore require extra pretraining or fine-tuning efforts on a customized

dataset. It follows from our previous literature review that the understanding of collective violence in multiple or mixed languages in social media remains underexplored.

Table 2.1. Summary of MLLMs and Twitter-based monolingual models. Adapted from Kalyan et al. (2021, tables 3 and 4) and Doddapaneni et al. (2021, table 1).

Model	Pretrained from	Pretrained tasks	Corpus	Lang.	Params	Vocabulary
mBERT	BERT	MLM, NSP	Wikipedia	104	172M	WordPiece (110K)
RoBERTa-Twitter	RoBERTa	MLM	Tweets (60M)	English	125M	SentencePiece (50K)
XLM-R	RoBERTa	MLM	CC-100	100	270M (base), 560M (large)	SentencePiece (250K)
XLM-T	XML-R	MLM	Tweets (198M)	30+	278M	SentencePiece (250K)
Bertweet-base	Scratch	MLM	Tweets (845M en + 5M COVID)	English	135M	WordPiece (64K)
Bertweet-large	Scratch	MLM	Tweets (873M)	English	355M	WordPiece (50K)
BertweetCovid19	Bertweet	MLM	COVID tweets (23M)	English	135M	WordPiece (64K)
CT-BERT	BERT	MLM, NSP	COVID tweets (23M)	English	335M	WordPiece (30.5K)
LaBSE	BERT	MLM, TLM	CC and Wikipedia (17B) + 6B translation pairs	109	471M	WordPiece (500K)
smaller-LaBSE	BERT	MLM, TLM	CC and Wikipedia (17B) + 6B translation pairs	15	219M	WordPiece (173K)

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

Our work builds on previous research conducted through the CODA Lab at NPS for the automated detection of hostile information campaigns using artificial neural networks. The researchers used a convolutional neural network widely used in computer vision for image segmentation problems called *U-NET* (Ronneberger et al. 2015). By dividing the earth's surface into approximately 20 million grid cells (5 km x 5 km), the model predicts the occurrence of violence in each individual cell on a weekly basis. The network is trained on seven different types of input data, including population density, roads, nighttime light emissions, conflict history, and a raw count of Twitter messages per grid cell, etc., each of which constitutes an input channel to the U-NET (Warren and Barreto 2020). Figure 3.1 shows a high-level overview of the model developed by the CODA Lab for predicting the occurrence of violent events at a global scale.

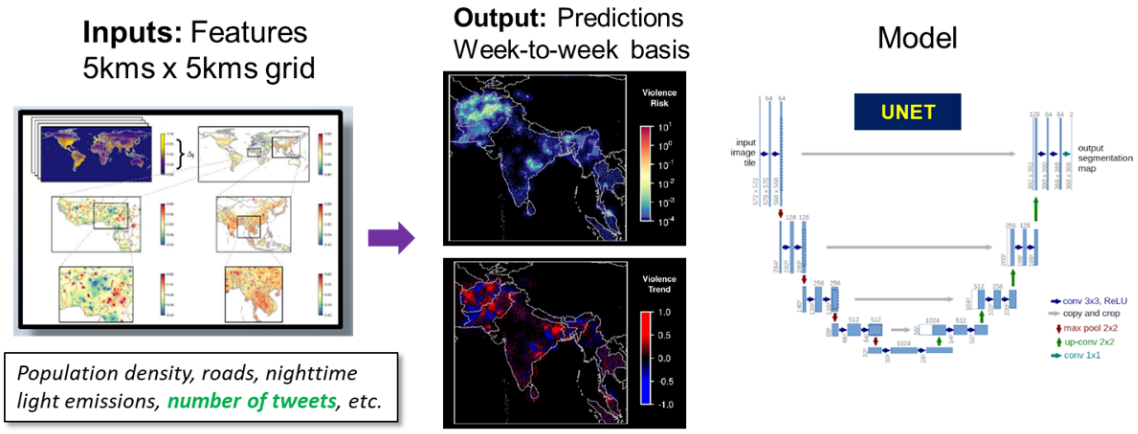

Figure 3.1. High-level overview of the violence prediction classifier on a global scale developed at the CODA Lab. Source: Warren and Barreto (2020).

The authors also found that the model observed a substantial decrease in predictive accuracy in the absence of social media inputs, which led them to support the claim that "social communication patterns are key to how the model successfully predicts future events of collective violence" (Warren and Barreto 2020, p. 8). With this acknowledged, our main contribution seeks to improve the predictive power of the raw Twitter counts fed into

the U-NET model by using state-of-the-art transformer MLLMs. We propose the four-step spatial-temporal framework illustrated in Figure 3.2 to achieve the desired objective. Furthermore, we hypothesize that there exists a sufficient collective violence signal in the Twitter discourse prior to the occurrence of violent events. The closer the conversation is to the violent events, in terms of time and distance, the stronger the presence of this signal. This hypothesis will drive the process of constructing our training corpus, which in turn will be validated on the basis of violent event prediction. If a predictive relationship is found between Twitter content and past events of collective violence in the world, this study could serve as a useful decision-aid tool for predicting violence on a global scale at the operational and strategic levels of war. The remainder of this chapter explains in detail each of the four constituents of our proposed framework.

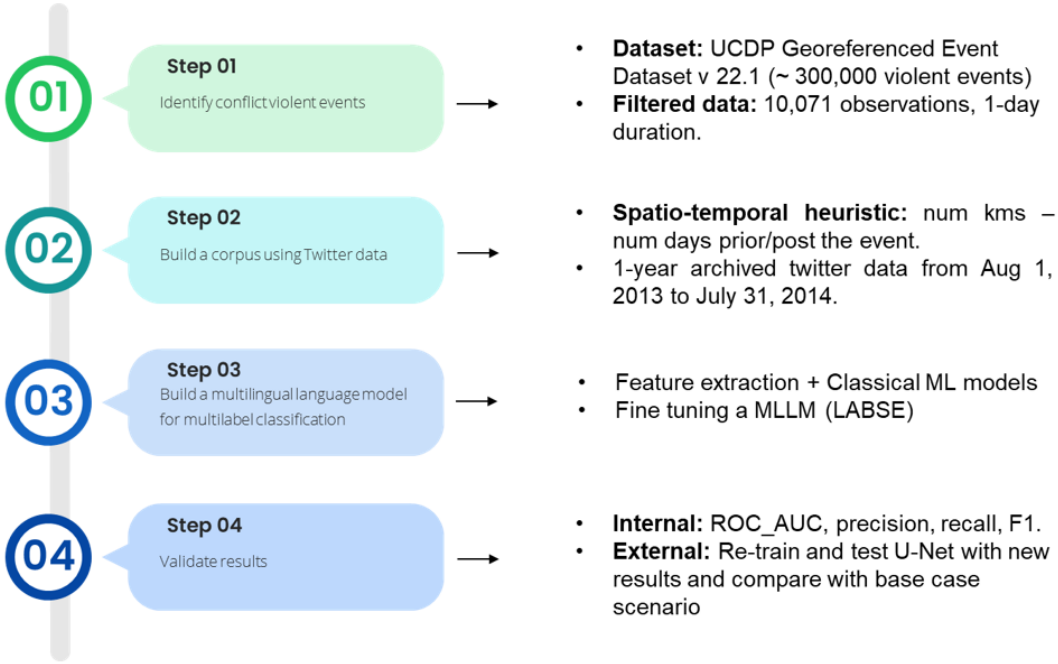

Figure 3.2. Four-step framework for building an end-to-end collective violence classifier. For step 2, we use the following values for our heuristics: num_kms = [10, 20, 30, 50, 70] and num_days = [1, 2, 3, 7].

The following discussion relies on the key notations summarized in Table 3.1.

Table 3.1. Key notations used in this thesis.

Notation	Description
χ	Feature space, which is M-dimensional
$\mathcal{Y}$	Label space
n	Number of training instances
m	Number of features
y_i	The ground-truth label
$\hat{y}_i$	The i^{th} prediction
(x_i, y_i)	The i^{th} labeled instance

3.1 Technical Problem Description

The data we aim to predict pose particular problems, because the violence signal contained in the training data is extremely sparse when compared to the signal of non-violent events. To address this shortcoming, we frame the problem as a multilabel classification problem. In this setting, each observation can belong to different spatial-temporal labels, allowing the model to pick up and amplify violence signals across different spatial-temporal scales. Formally, a multilabel classification problems is defined as:

> Given a training set, S = $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $1 \leq i \leq n$, consisting of n training instances, $(\boldsymbol{x}_i \in \chi, y_i \in \mathcal{Y})$ i.i.d drawn from an unknown distribution D, the goal of multi-label learning is to produce a multi-label classifier h : $\chi \rightarrow \mathcal{Y}$ (in other words, h : $\chi \rightarrow 2^L$) that optimizes some specific evaluation function (i.e., loss function) (Sorower 2010, p. 3).

Typically, classification is a task that assigns a single label to each instance in our dataset. In a binary setting, there are only two possible target labels. Alternatively, the classifier might involve predicting more than two labels (multiclass setting). In both settings, the target labels are mutually exclusive, meaning that the instance belongs to one class only. However, there is another type of problem, multilabel classification, that involves predicting more than

one class label for each observation; here the class membership is not mutually exclusive. Multilabel classification problems can essentially be broken down into sets of binary mini problems without much loss of information. Figure 3.3 illustrates the differences between binary, multiclass, and multilabel classification. In the binary and multiclass setting, the prediction probabilities must add up to one, whereas in a multilabel classifier this constraint does not hold true.

Figure 3.3. Binary vs. multiclass vs. multilabel classification. For binary and multiclass classification the labels belong to one class only, whereas for multilabel classification the class membership is not mutually exclusive. Source: Arghyadeep (2021).

Multilabel classification is a topic that is barely touched upon in many ML libraries. In some cases, we need to write most of the code for certain tasks, and the evaluation metrics require either third-party specialized libraries or some workarounds for deriving meaningful results. In contrast, transformer models natively provide support for multilabel classification problems just by indicating the number of class labels as nodes in the output layer. For example, if our task requires six target labels, then the transformer model will expect six nodes in the output layer. In each node, a sigmoid activation function will predict the probability of class membership for that particular label. During training, the model is fit with a *Binary Cross-Entropy* loss function.

Some transformer models can perform multilabel classification out of the box by specifying an argument during model instantiation. This ensures that the loss function used in the forward pass is suitable for multilabel tasks. However, not all transformer models provide

this support, in which case we are required to override the forward method of the base-model's main class ourselves. This extra step allows us to compute the loss with a sigmoid instead of softmax function applied to the logits. Fortunately, the transformer models we are using in this research belong to the former type.

3.2 Identifying Collective Violence Events

The starting point of this research is the Uppsala Conflict Data Program (UCDP) dataset, which provides a collection of georeferenced organized violence events in the post-1989 world (Sundberg and Melander 2013). The version used in our project is the *GEDEvent_v22_1* dataset, containing approximately 300,000 records with more than 40 features (e.g., date, latitude, longitude, country, city, province). The basic unit of analysis in the dataset is the *event*, defined as "An incident where armed force was used by an organised actor against another organized actor, or against civilians, resulting in at least 1 direct death at a specific location and a specific date" (Högbladh 2022, p. 4). For this research, we limit our analysis to the one-year period from August 1, 2013, to July 31, 2014, to match the timespan of our secondary data source (more to follow in the next section). Moreover, we restrict the conflict duration to one day or less under the hypothesis that during this short period of time the change in discourse between pre-violence and post-violence conversations is more evident. The raw observations hitherto correspond to 24,132 events over the one-year timeframe. Finally, we further filtered the dataset according to the below criteria, yielding a total of 10,071 unique records. For comparison purposes, Table 3.2 details the distribution of violence events per region in the world before and after filtering the raw dataset.

- Types of violence: State-base conflict, non-state conflict, and one-sided violence. The three events are mutually exclusive.
- A geo-precision of 1, whose value indicates that an "event can be related to an exact location, meaning a place name with a specific pair of latitude and longitude coordinates" (Högbladh 2022, p. 21).
- A temporal precision value of 1 meaning that "the exact day of the event is known" (Högbladh 2022, p. 23).
- Remove spatio-temporal duplicates.

Table 3.2. Distribution of violent events per region from August 1, 2013, to July 31, 2014.

Data Set Dist.	Africa	Americas	Asia	Europe	Middle East	Total
Unfiltered data set	2,096	245	2,933	342	18,516	24,132
Filtered data set	917	100	877	94	8,083	10,071

Figure 3.4 illustrates the density of the 10,071 unique violence events within our analysis timeframe. Notwithstanding the highest patterns of violence appear more concentrated in the Middle East, Asia, and Africa, this visual illustration depicts a wide diversity of locations, implying broad coverage of the many languages spoken in the world. This language diversity is what motivates the use of MLLMs in our research. Each violent event in the final dataset is a unique combination of date and georeferenced location, which serves as a lookup table for building the Twitter corpus in the next section.

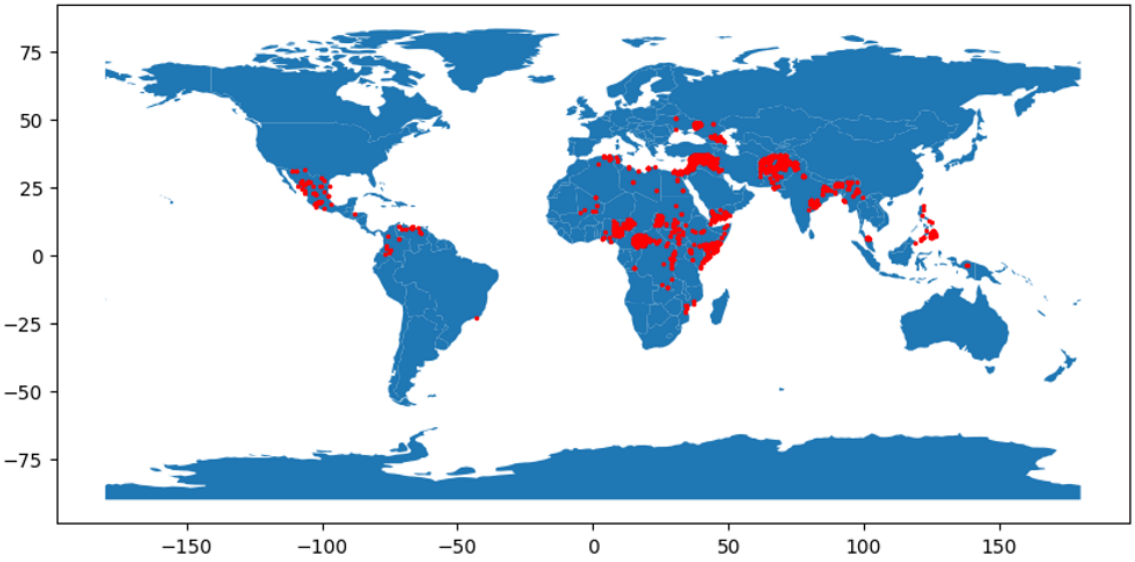

Figure 3.4. Map of violent events from August 1, 2013, to July 31, 2014. The red dots correspond to 10,071 geolocated unique violent events. Source: Sundberg and Melander (2013).

3.3 Building the Twitter Multilingual Corpus

We built the Twitter multilingual corpus by comparing the previously described list of collective violence events from UCDP and a historical archive of Twitter messages licensed for research at NPS. This archive comprises a randomized sample of 10% of Twitter traffic from August 1, 2013, through July 31, 2014, whose size is approximately 40 terabytes, recording approximately 12 billion separate Twitter messages. To allow for proper geospatial analysis, we used a spatial-temporal heuristic and grid approach to efficiently calculate the geodesic distances between the available tweets in the archived dataset and the list of violence events. This approach is close in spirit to the procedure used in a related master's thesis at NPS for understanding violence through social media in Iraq (Frost et al. 2017). In that work, the authors also used both datasets, UCDP and the Twitter archived data from NPS, but to extract Twitter metadata from Arabic tweets as opposed to Twitter content in all languages. For what follows, we will explain in detail the logic used in the code for building the multilingual corpus as illustrated in Figure 3.5.

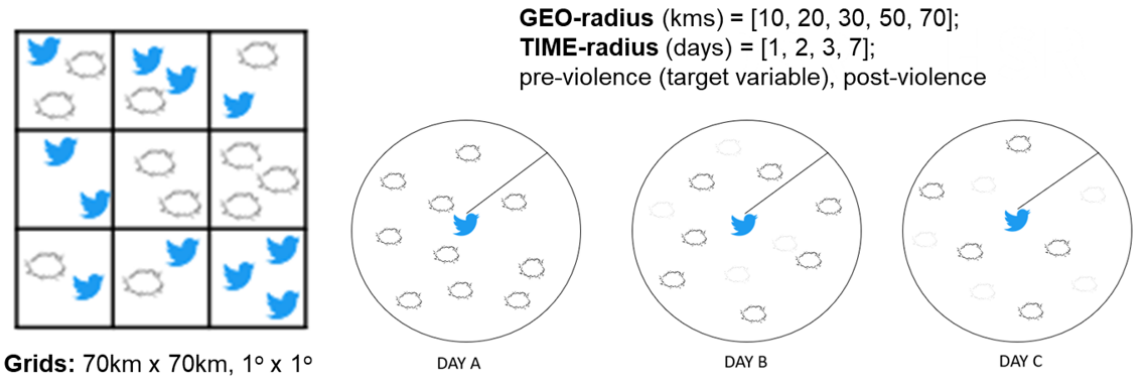

Figure 3.5. Illustration of the spatial-temporal heuristic used for building the Twitter corpus. The different combinations of spatial-temporal heuristics yield a total of 40 different labels.

The first step is to create a global grid of Twitter messages where each 1 degree by 1 degree latitude/longitude grid cell can be rapidly queried for a specific date (see the left graphic of Figure 3.5). Some basic parameters are set, including the start and end dates for the queries, overall latitude/longitude bounds for each grid, and any metadata fields of interest that will be retained from each Twitter record. Most importantly, *TIME_RADIUS* sets the number of days that will be used as thresholds for the pre/post periods around each conflict event,

and *GEO_RADIUS* sets the number of kilometers that will be used as thresholds for linking tweets to conflict events. Note that multiple values are included for each tweet, because the procedure implemented here allows us to assign multiple tags to each message (multilabel classification), with each tag representing a different combination of space/time thresholds.

Once the grid is defined, we then load into a shared memory database all the georeferenced Twitter records from the year-long archive, as well as the unique location-date observations from the conflict event dataset (matching the gridded structure of the Twitter database). The core idea of this logic is to efficiently calculate the geodesic distances between the coordinates of each tweet and the coordinates of each georeferenced violence event per each grid cell (see the right graphic of Figure 3.5). A naive approach would be to calculate all these pairwise distances between the billions of tweets contained in the archived dataset and the 10,000 records of the list of violent events. This would simply be a computationally prohibitive operation, which calls for a more sophisticated approach such as the gridded structure used in our work.

The core logic in our approach takes a longitude-latitude-date query key (defining the focal cell) and pulls all available Twitter records for that location and date. It then loops over the conflict events associated with that cell and its neighbors, and calculates the geospatial distance between each tweet and each conflict event in this neighborhood. This calculation is expanded to the eight neighbors of the focal grid cell (1 lat/lon degree in each direction) to account for the events near the grid boundaries. It then loops over the values in *GEO_RADIUS*, and loops over the values in *TIME_RADIUS*, and loops over the *pre* and *post* periods, and creates a separate label for each combination. If any event is found within both a given spatial radius and a given temporal radius, this label is coded as 1; otherwise, it is coded as zero.

All tweets with a code of 1 for at least one label are retained and saved in a CSV file along with the labels and other associated metadata. Each label refers to a particular spatial-temporal window. For instance, the label *pre7geo30* will equal 1 if a conflict event was found within the seven days following the tweet and within 30 kilometers of the location of the tweet, and the label *post2geo50* will equal 1 if an event was found within the two days prior to the tweet and within 50 kilometers of the location of the tweet. We included temporal radius values of 1, 2, 3, and 7 days, and spatial radius values of 10, 20, 30, 50, and

70 kilometers, yielding a total of 40 different labels for each tweet. The resulting dataset is stored in a relatively large CSV file, 2 GB zipped, and 9 GB when unpacked, with labels for 23,291,575 tweets in total. This raw dataset is then further pre-processed to remove trailing spaces, urls, mentions, and retweets. Finally, we split the data into 90%-10% train-test splits, and the training split is further split into 80%-20% train-validation splits. Table 3.3 summarizes the number of observations in each of the three splits.

Table 3.3. Number of observations in each dataset split.

	Train	**Validation**	**Test**	**Total**
Num. observations	16,769,932	4,192,483	2,329,158	23,291,573

On Twitter and some other social media platforms, users must explicitly enable the GPS coordinates associated with their messages. For example, GPS coordinates are only available for 1% of the active Twitter accounts; hence, we expect to find the same proportion in the archived data. To overcome this limitation, we used the profile-based coordinates provided in the licensed archive, which correspond to the geo-location of the hometown reported in each public user profile. Although these coordinates are not as accurate as the GPS coordinates, using this approach we increased the volume of data available for building the corpus "the profile-based geo-coordinates are available for approximately 30 % of the records in the NPS archive" (Frost et al. 2017, p. 18). All the pairwise distance calculations were computed in a 3 TB RAM server, using a distributed, in-memory database developed in the CODA lab called *KVswarm*. The time it takes to construct the corpus is approximately six hours, as shown in Table 3.4

Table 3.4. Computation performance during corpus construction.

Task	**Time (hrs)**
Load the archived Twitter data into shared memory database	5.25
Parallel execution of the corpus construction routine	0.5

As it was previously stated in Chapter 1, one of our main contributions in this research is to make the dataset publicly available. Unfortunately, sharing the dataset "as is" would result in a clear violation of the license agreement for the proper use of the archived data. This agreement forbids the sharing of the actual Twitter content (text), but not the numeric tweet identifiers. Users wishing to use our dataset will need to rely on the Twitter application programming interface (API) to query the Twitter database using the *tweet ID* attribute in order to retrieve the original Twitter *text* for each message. A sample of the proposed dataset is detailed in Table 3.5, where each row corresponds to a unique *tweet ID* with the six different labels used in our work.

Table 3.5. Schema of the dataset that will be made publicly available. Each row corresponds to a unique *tweet ID* with the associated six different labels used in our current work

.

Tweet ID	pre7geo10	pre7geo30	pre7geo50	post7geo10	post7geo30	post7geo50
tweet_ID1	0	1	1	0	0	1
tweet_ID2	1	1	1	0	0	0
tweet_ID3	0	0	1	0	1	1

3.4 Training a Multilingual Language Model

Multilingual NLP systems are designed to handle user input in more than one human language. There are a couple of ways to achieve this objective. One way is to create separate subsystems for each language of interest. In theory, this approach is significantly easier because it allows for the use of pre-built systems and components readily available in the NLP community. However, there is another other option that requires designing a single system, an MLLM that can handle inputs in multiple or mixed languages. This is the approach used in our work. In general, MLLMs are used in situations where the system will be used by different language populations. Predicting worldwide collective violence seems to follow this intuition. A high-level overview of how transformer models are trained is illustrated in Figure 3.6. These steps are covered in greater detail in the remainder of this section.

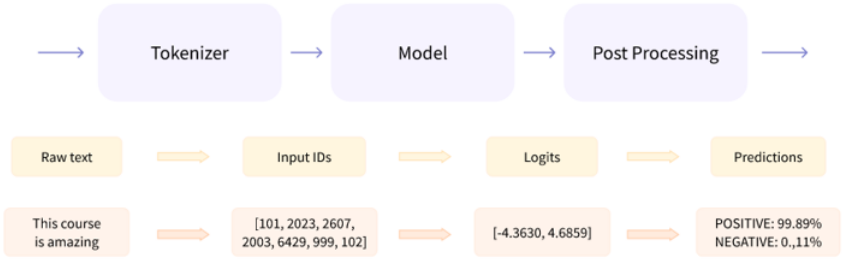

Figure 3.6. A typical pipeline for training transformer models. The raw text is first tokenized into numbers that the model understands. Given the input tokens, the model outputs logits, which are post-processed into probabilities. Source: Abid et al. (2022, Ch. 2-2).

3.4.1 The Transformer Architecture

The model block in Figure 3.6 refers to the transformer architecture, or language model, used in the pipeline. Our work uses three different transformer models: XLM-T, LaBSE, and smaller-LaBSE (refer to Table 2.1 for a thorough description of the main features of these models). To understand how these models work, we need to trace their origins to the original transformer architecture and a language model called BERT.

The transformer architecture is a deep neural network model proposed in 2017 by a team of Google researchers in a seminal paper called *Attention is all you need* (Vaswani et al. 2017). The model is an encoder-decoder architecture originally designed for translation, as illustrated in Figure 3.7. The output of the encoder is fed as an input to the decoder, which produces the desired sequence in the target language. The main components of the vanilla transformer architecture are defined as follows:

- *Encoder.* This maps an input sequence of tokens to a sequence of continuous representations, or embedding vectors, also known as hidden states or context. In short, it creates an embedding for each word based on the relevance, or context, to other words in the sequence.
- *Decoder.* This receives the output of the encoder (hidden state) together with the decoder output at the previous step to iteratively generate an output sequence, one step at a time.
- *Attention layers.* These allow the model to pay specific attention to the most important

words in the sequence. In doing so, the model discovers more relevant semantic and syntactic information. This attention mechanism allows the transformer architecture to process text in both directions, hence the name bidirectional.

- *Attention mask.* This prevents the model from paying attention to particular words in the sequence.
- *Positional Embeddings.* These contain positional information for each token to account for its relative position in the sequence.

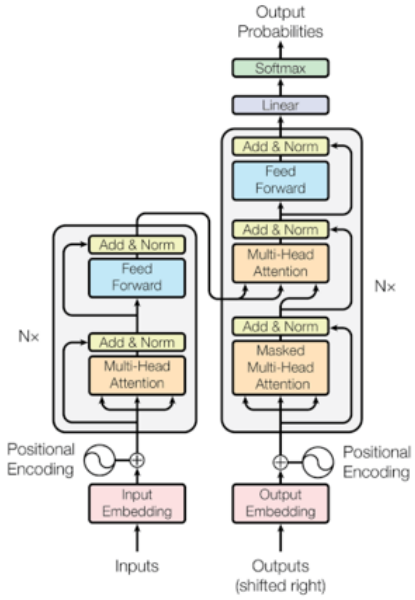

Figure 3.7. The transformer architecture as described in the *Attention is all you need* paper. The original implementation uses an encoder (left) and a decoder (right) component in tandem. Each colored block corresponds to a different layer in the architecture. Source: Vaswani et al. (2017, p. 3).

BERT, on the other hand, is a language model that is a derivation from the original transformer architecture. We can extract some key insights about how BERT works just by looking at its full name *Bidirectional Encoder Representations for Transformer Models*. First, BERT is based on the transformer architecture. Second, BERT uses only the encoder part of that architecture; therefore, its output is an embedding and not a textual word, which makes it unsuited for tasks like text generation or translation. Lastly, BERT is bi-directional, which means that it learns information from both the left and right side of the sequence during training. According to the original paper, BERT was trained using Wikipedia text,

WordPiece tokenization, and two pretraining objectives MLM and NSP (Devlin et al. 2018).

Another model, XLM-T, which is multilingual, is pretrained from a base model dubbed XLM-R using CPT (Barbieri et al. 2022). XLM-R in turn is based on the RoBERTa model—a member of the BERT family of transformers— pretrained on monolingual data using a TLM objective (Lample and Conneau 2019). As such, XLM-T inherits all the benefits from a TLM pretraining objective; however, its training corpus is composed of 198 million tweets in 30 different languages (Barbieri et al. 2022). Although its roots can be traced to BERT, XLM-T uses a different tokenization scheme (*SentencePiece*) with a vocabulary of 250,000 subword tokens (Barbieri et al. 2022). XLM-T is the only MLLM whose framework specifically addresses Twitter content, being the only available model in the Hugging Face Hub trained on multilingual Twitter data. This unique feature makes XLM-T a good starting point for our research project.

An MLLM developed by researchers at Google AI, LaBSE is an extension of mBERT, a multilingual version of BERT, that produces "language-agnostic sentence embeddings for 109 languages" (Feng et al. 2020, p. 1). A careful examination of its full name, *Language-Agnostic BERT Sentence Embedding*, tells us that its architecture resembles closely the original BERT architecture discussed previously. Similar to mBERT, LaBSE is pretrained using MLM and TLM on a corpus of 17 billion monolingual sentences and six billion bilingual sentence pairs, yielding a vocabulary with 500,000 subword tokens generated using *WordPiece* tokenization (Feng et al. 2020). The resulting model performs particularly well even on low-resource languages not seen during training.

At a high level, LaBSE uses a dual-encoder architecture, where each encoder is initialized with a set of weights from an already pretrained mBERT model (see Figure 3.8). Since LaBSE uses bilingual sentence pairs for training, a sentence in a source language passes through the first encoder while the sentence in the target language goes to the second encoder. The model takes both vectors, encoded separately, for calculating a similarity loss to measure the performance of the translations. The loss is fed back to the transformer model to update its weights.

Although the model uses two different encoders, the fact that both blocks share the same weights means that we are effectively updating a single model. This technique allows

the model to learn something general across different languages. First, during the initial pretraining phase, the model learns general patterns from all sorts of texts and not necessarily from which language the text is coming from. Then, during the second phase with the two encoders, the model is actively learning labels related to translations. In short, given that there is enough data and enough training time available, the model should be able to come up with language-agnostic representations. To have an idea of the amount of training time required to achieve this level of performance, the authors from Google used 1.8 million epochs during training (Feng et al. 2020). Fortunately for us, we can use this pretrained model and fine-tune it to our specific task using the techniques described next.

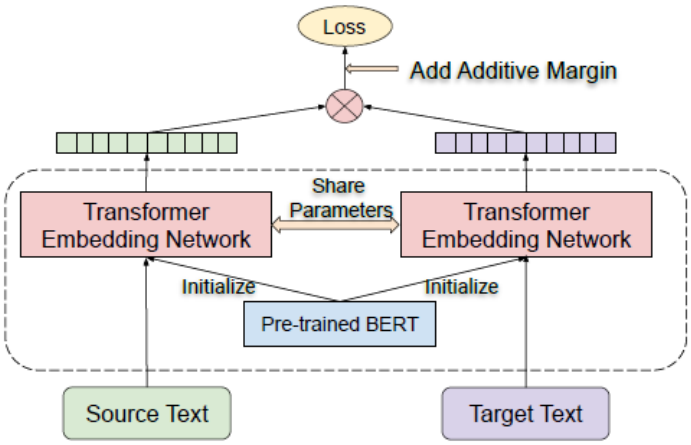

Figure 3.8. Dual-encoder architecture of LaBSE. The model learns language-agnostic embeddings by feeding parallel data to the dual encoders. Input text in one language is fed to one encoder, while text in the target language is fed to the second encoder. Source: Feng et al. (2020, p. 1).

3.4.2 Traditional ML Multilabel Classifiers

Model building in a multilabel scenario is different from the traditional approach in binary or multiclass classification. Multilabel classification is a problem that requires advanced methods and specialized machine learning algorithms to make predictions. There is no constraint on how many target classes we assign in a multilabel setting; however, the larger the number of labels, the more complex the problem becomes. The three most widely used techniques to cope with multilabel problems are problem adaptation, problem transformation, and ensemble methods. These techniques are implemented in code using popular

libraries such as scikit-learn (Pedregosa et al. 2011) and scikit-multilearn (Szymański and Kajdanowicz 2017).

Problem transformation refers to the process of converting a multilabel dataset into a single-labeled dataset where the existing binary classifiers can be applied (Sorower 2010, p. 3). This strategy in turn is further subdivided into three methods: binary relevance, classifier chains, and label powerset. Let us illustrate how these methods work using the following setting. Given an input vector χ of size (n x m) and four different labels ($y_i \epsilon \mathcal{Y}$; for i = 1, 2, 3, 4), we find a multilabel classifier $h : \chi \rightarrow \mathcal{Y}$ using the problem transformation strategy as follows:

- In *binary relevance*, a single-label binary classifier is trained independently on the original dataset to predict each class membership (Sorower 2010, p. 4). In our example, the problem is split into four classifiers (χ, y_1), (χ, y_2), (χ, y_3), and (χ, y_4); therefore, four outputs are generated. One potential drawback of binary relevance is that we may lose label correlation, which is addressed in the next two methods.
- *Label powerset* takes into account possible correlations between class labels. It transforms the problem into a multi-class classification problem by mapping each combination of labels into a single label and then training a single-label classifier (Sorower 2010, p. 4). In the previous example, there are 2^4 distinct label combinations; therefore, the problem takes the form $h : \chi \rightarrow \mathcal{Y}$; for $\mathcal{Y} = 0000, 0001, \ldots, 1111$. One potential drawback of this method is that as the number of classes increases, the distinct combinations grow exponentially. This is even more evident if the dataset is large like the one used in our work.
- In *classifier chains*, multiple classifiers are connected to a chain in a sequential process, where the output of one classifier is the input of the next one (Read et al. 2011). The first classifier is constructed with the input data and the first label (χ, y_1). The second classifier takes the first classifier as input and the second label as output $((\chi, y_1), y_2)$. The process continues until we get the four possible combinations.

Problem adaptation uses algorithms that inherently handle multilabel classification rather than transforming the problem into mini classifiers. Scikit-learn (Pedregosa et al. 2011, see the multiclass and multioutput algorithms section) and scikit-multilearn (Szymański and Kajdanowicz 2017, see section 5) provide a small handful of these algorithms (i.e., decision

trees (DT) (Breiman et al. 1984), K-nearest neighbors (KNN) (Hastie et al. 2009, p. 463), Multilayer Perceptron (Hastie et al. 2009, p. 130), RF (Breiman 2001), Extra Tree (Geurts et al. 2006), DT (Breiman et al. 1984), and Ridge (Hastie et al. 2009, p. 61)). In contrast to problem transformation, these algorithms do not use a wrapper when implemented in code.

Ensemble methods are a hybrid technique that combines the functionalities of problem adaptation and problem transformation. Intuitively, ensemble methods combine multiple algorithms into a single model, which typically performs better than the two other approaches. The confidence estimates and the aggregation scheme are key factors of these ensembles. Models such as XGBoost (Chen and Guestrin 2016), Bagging (Breiman 1996), Boosting (Friedman 2001), and RF (Breiman 2001) fall under this category.

3.5 Evaluating a Multilingual Language Model

The metrics we choose to evaluate machine learning models influence how the performance of these algorithms is measured and compared. Evaluation refers to the process of measuring how far the model predictions are from the actual ground truth labels. For traditional classification such as multiclass problems, there exists a set of standard metrics that can be used to evaluate the output predictions. However, none of these metrics can be applied in their original form in a multilabel setting, because the latter implies a notion of being *partially correct* (Sorower 2010, p. 13). Unlike normal classification tasks where the target labels are mutually exclusive, multilabel classification requires the application of metrics to each target separately.

For what follows, we will restrict our analysis to the following evaluation metrics: accuracy, precision, recall, F1, and ROC-AUC. To illustrate how these metrics work, we will refer to an example inspired by Karakaya (2020), as depicted in Figure 3.9. The example assumes that we have three instances with four different labels: A, B, C, and D. The model outputs some losses at the end of the training process, which are then passed through a sigmoid function in order to generate probabilities. We generate predictions by comparing those probabilities to a specified threshold (0.5 in this case).

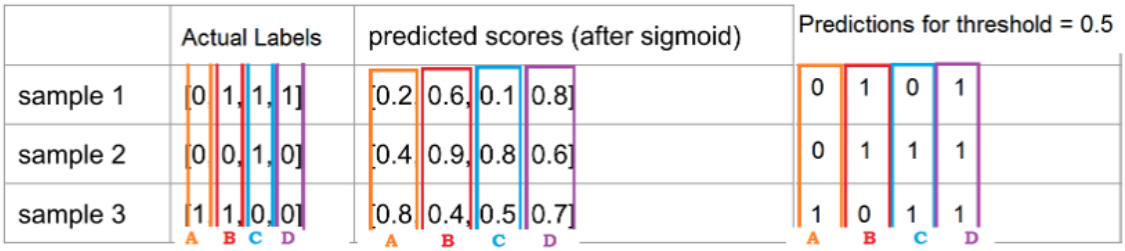

Figure 3.9. Output predictions in a multilabel classifier. Source: Karakaya (2020).

3.5.1 Accuracy

Accuracy is the fraction of correctly predicted labels in the dataset (see Eq. (3.1)). For this metric, the set of labels predicted for a sample must strictly match the corresponding set of ground-truth labels. This definition requires that the four labels of each sample in Figure 3.9 exactly match the 4-class predictions to be correct (output = 1). Since there are no matches, the overall accuracy is zero. This metric is not suited for multilabel classification, because it is not label-based.

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|}, \tag{3.1}$$

3.5.2 Precision, Recall, F1 Score

In multilabel classification, these three metrics are applied to each label independently. An intuitive way to understand these three metrics is by using a confusion matrix from a typical classification report, as illustrated in Figure 3.10. Scikit-learn provides few ways to combine results across labels by using the *average* argument (required for multiclass/multilabel targets) in the corresponding score functions. This argument can take five different values, but we will restrict the definitions to the two values used in our research:

- *Micro*: Calculate metrics globally by counting the total true positives, false negatives and false positives.
- *Weighted*: Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). (Pedregosa et al.

2011, see the metrics module).

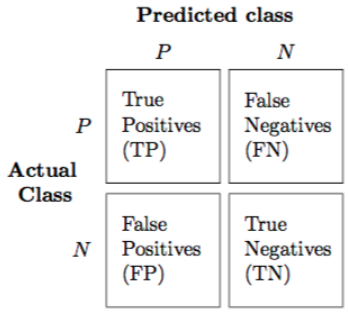

Figure 3.10. Confusion matrix from a classification report.

Precision is the proportion of true positives to the total number of positive predictions (see Eq. (3.2)). In the example in Figure 3.9, a *micro-precision score* is calculated by counting the number of true positives (TP)'s and false positives (FP)'s for each label. In this case, TP = 4 (i.e., second column of actual labels and predictions for sample 1) and FP = 4 (i.e., third column of predictions and actual labels for sample 3), which yields a micro-precision score of 0.5.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}. \tag{3.2}$$

Recall is the proportion of true positives to the total number of positive instances (see Eq. (3.3)). It is also known as *sensitivity* or the true positive rate (TPR). Following the logic in our previous calculations, the number of false negatives (FN) = 2 (i.e., third column of actual labels and predictions in sample 1); therefore, the micro-recall score is 0.67.

$$Recall(Sensitivity/\text{TPR}) = \frac{TruePositives}{TruePositives + FalseNegatives}. \tag{3.3}$$

F1 score is defined as the harmonic mean of precision and recall (see Eq. (3.4)). Calculating the micro-f1 score for our example is simply a matter of replacing the previously computed values of micro-precision and micro-recall in Eq. (3.4) for a score of 0.57.

$$F1 = 2 * \frac{precision * recall}{precision + recall}. \tag{3.4}$$

3.5.3 Receiver Operating Characteristic Curve (ROC) and AUC Score

A receiver operating characteristic curve (ROC) summarizes the trade-off between the TPR and the false positive rate (FPR) for a predictive model using different probability thresholds. In other words, it plots the false alarm rate versus the hit rate. This is a useful tool when predicting the probability of a binary outcome. TPR is known as *sensitivity* or *recall*, and FPR as one minus *specificity* (see Eq. (3.5)). The AUC score, given by the area under the ROC curve, summarizes the model skill in just one number between 0 and 1. In a multilabel setting, the *ROC-AUC score* is calculated by averaging over all labels, and the *ROC curve* is plotted for each label. Figure 3.11 illustrates an ROC curve for a toy multi-label example.

$$\text{FPR} = \frac{FalsePositives}{FalsePositives + TrueNegatives}. \tag{3.5}$$

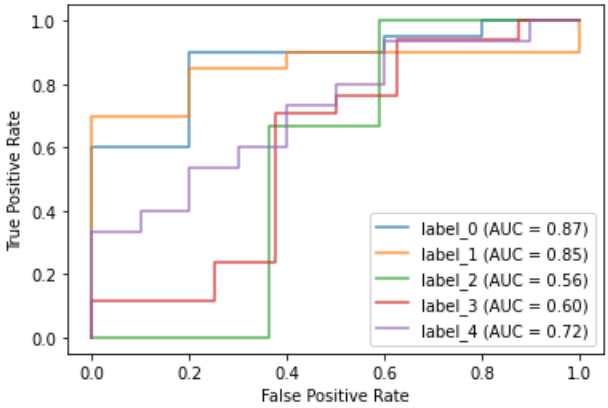

Figure 3.11. ROC curve for multilabel classification. In this setting, a ROC is plotted for each label. Source: Karakaya (2020).

When working with large transformer models, the same metrics described thus far can be calculated using the *Evaluate API* available at the Hugging Face Hub. There are numerous advantages for using this API instead of scikit-learn when evaluating our models:

- The metrics can be calculated with the entire dataset, in batches, or in a distributed fashion.
- They are optimized to work seamlessly with the rest of the libraries available in the Hugging Face Hub (i.e., transformers, datasets, accelerate).
- We can bundle together different metrics (i.e., F1, precision, recall, ROC-AUC) using the *combine()* function to capture different aspects of the model in just one call. It is more efficient than calling the metrics sequentially.
- The results can be pushed to the hub for saving and sharing.

CHAPTER 4:
Experimentation and Results

We started our work by asking the research question: "To what extent can the information contained in social media (Twitter) generate sufficient signals to efficiently predict collective violence events using multilingual transformer models?" We attempt to answer this question by comparing the performance of traditional ML classifiers against MLLMs. In both cases, we use three MLLMs (i.e., smaller-LaBSE, LaBSE, and XLM-T) to either extract features from our data or to serve as end-to-end classifiers. In the former case, we trained five different ML classifiers. We carefully chose the five models to represent the three most widely used techniques for multilabel classification: problem adaptation, problem transformation, and ensemble models. Likewise, for the deep learning approach, we fine-tuned smaller-LaBSE, LaBSE, and XLM-T using pre-trained checkpoints available in the Hugging Face Hub. The results produced four noteworthy findings described in the remainder of this section. But before we dive into our main findings, let us start by performing an exploratory data analysis.

4.1 Exploratory Data Analysis

Every data science project starts with an exploratory data analysis (EDA) for summarizing the main characteristics of the dataset. The size of our dataset is roughly 23 million tweets split into three datasets: training, validation, and testing sets (see Table 3.3). The raw text is further pre-processed by stripping extra white spaces and removing url's, retweets (RT), and mentions (@username). This data cleaning is required before entering the next step in the classification pipeline.

Figure 4.1 shows the class distribution for the 40 different label combinations from the spatial-temporal heuristics described in Section 3.3. A careful examination of Figure 4.1 demonstrates two distinct observations. First, the amount of data available in our corpus is inversely proportional to the distance from where the tweet originates to the location of the violent event. For example, if we compare *pre1geo10*—the closest spatial-temporal heuristic with one day and 10 kms—to *pre7geo70*—the furthest heuristic with seven days and 70 kms—we note that the former is about six times rarer. This behavior is somewhat

intuitive because we would expect to have less Twitter traffic in smaller geographical areas and time windows. This evidence suggests that if we want to build a classifier with distant spatial-temporal heuristics, then we would expect to have a heavily imbalanced dataset, which in turn would require a different treatment when dealing with evaluation metrics and training losses (Tunstall et al. 2022, p. 27). In the case of our project, the distribution of observations across the six chosen target labels is sufficiently balanced.

Second, Twitter traffic for similar heuristics—before and after the occurrence of a violent event—remains largely unchanged. For example, if we compare the labels *pre3geo20*—which corresponds to three days and 20 kms for pre-violence—and respectively *post3geo20* for post-violence, we note that the length of the horizontal bars in the graph are similar. This observation is best exemplified in the left picture of Figure 4.2 for two target labels used in this project. This behavior could be considered counterintuitive, because we might expect an increase in social media traffic where people are expressing their discomfort and disapproval after these atrocities. If traffic is not increasing in the aggregate, this may imply that the discourse in Twitter shifts drastically to violent conversations once a violent event takes place, or it may indicate that most of the conversations are unrelated to the nearby violent events. This is a question that deserves further exploration.

Finally, there is a significant byproduct of non-violence observations after applying the spatial-temporal heuristics in the dataset, even though non-violence was not treated as a separate target class. Recall that the two major class categories in the dataset are pre-violence and post-violence, each with a different subset of labels. A non-violence instance refers to an observation whose labels are all equal to zero. The right picture of Figure 4.2 shows the distribution of non-violence (all labels equal to zero) and violence observations (at least one label is equal to one) for the classes *pre7geo30* and *post7geo30*. Note that there is a significant amount of non-violence observations within these two labels, it is roughly equivalent to one third of the dataset. This is important because all the classifiers will learn non-violence patterns during training. Then, at inference time, any predictions whose labels are all equal to zero should be interpreted as non-violence events, which are indeed associated with a corresponding pattern of non-violence learned by the model.

Transformer models are well suited for sequence data and for learning long-term dependencies; however, these nice features are limited to the maximum input sequence length

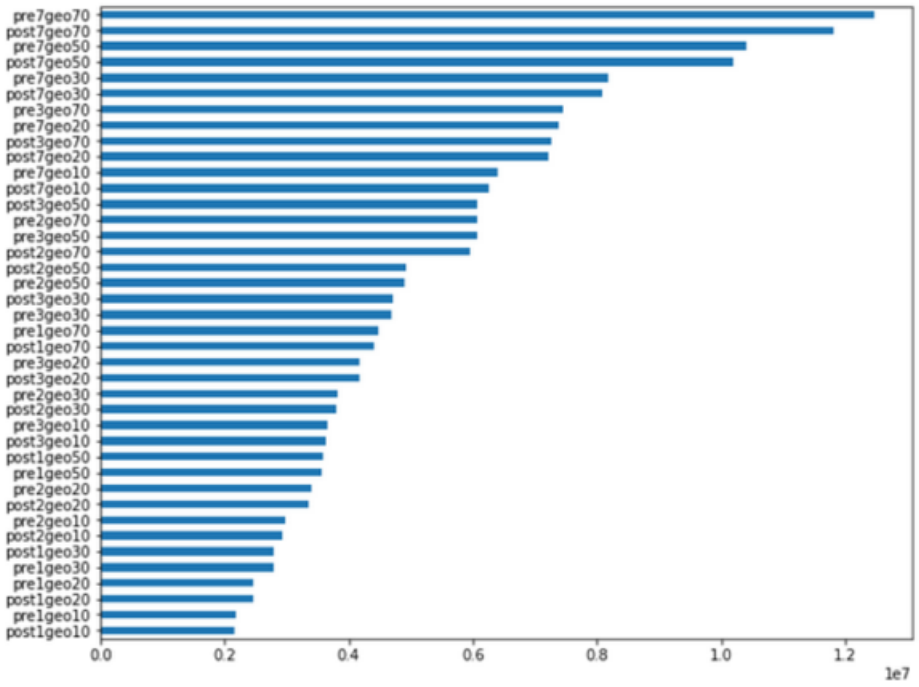

Figure 4.1. Class distribution in the Twitter corpus.

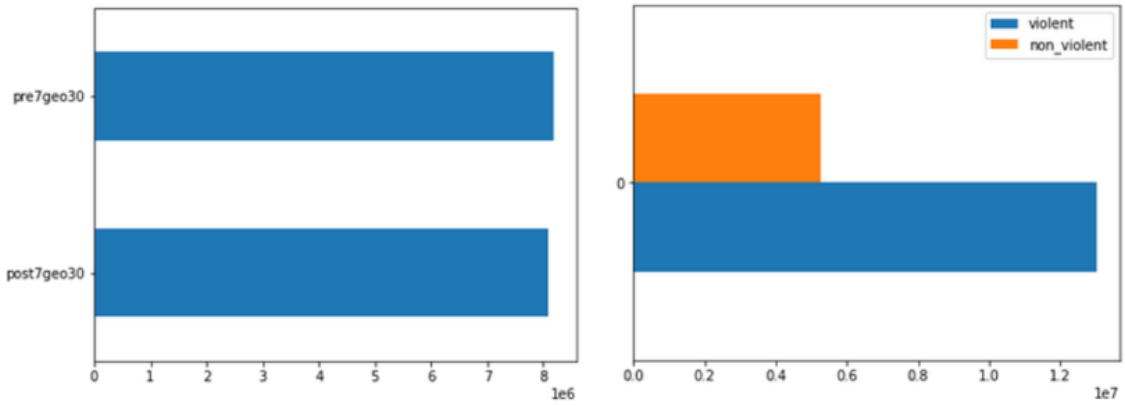

Figure 4.2. Visualization of a sample of the balanced dataset. (Left) Both labels *pre7geo30* and *post7geo30* are balanced. (Right) Comparison of violence and non-violence data. For illustration purposes, non-violence data refers to having all labels equal to zero.

the models can handle, also known as maximum context size (Tunstall et al. 2022, p. 28). LaBSE, smaller-LaBSE, and XLM-T, all descendants of the original BERT architecture,

can handle input sequences of 512 tokens. In the world of transformers, longer sequences are truncated and shorter sequences are padded with zeros to achieve the fixed-length batches the models expect during training. Moreover, models with longer input sequence lengths are more expensive to train because we are feeding more data to the network. Since each tweet is limited to 280 characters, it makes sense to use shorter context sizes when training transformer models. For what pertains here, Figure 4.3 shows a rough estimate of the tweet lengths (number of words per tweet) in the dataset. Note that in both box plots, the minimum length is 1, whereas the maximum length is 32. Based on this evidence, we set the sequence length for all the MLLMs equal to 32, thus avoiding unnecessary computational costs.

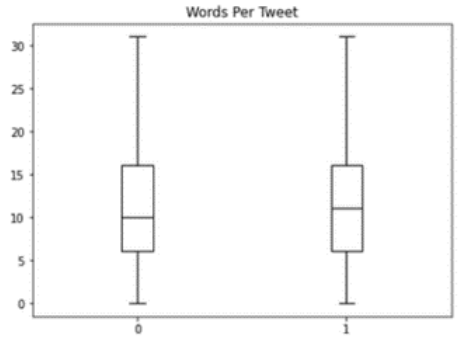

Figure 4.3. Distribution of words per tweets.

Lastly, any multilingual application would first and foremost require knowing the number of languages it intends to address. Figure 4.4 shows the distribution of the 68 languages existing in our corpus, including the *und* or undefined language category (see the top picture of Figure 4.4), as categorized by Twitter. Every time Twitter fails to identify the language of a tweet, it assigns a label *und*; this could happen because the message is too short, or it does not contain any text, only but images or emojis. The bottom-left picture of Figure 4.4 shows the language distribution of our corpus. The plot shows a heavy-tailed distribution, where most observations are contained within the first few high-resource languages, and the remaining low-resource languages lie in the long tail. A better visualization of this distribution in log scale is illustrated in the bottom-right plot of Figure 4.4, where we observe that the last two languages have extremely scarce data instances. This evidence suggests that perhaps an MLLM which targets 20 or so languages might be enough to approach our problem at hand, an observation that deserves more rigorous analysis.

The language distribution shown in Figure 4.4 is a good approximation of the predominant

languages in the Twitter discourse around the world. Surprisingly, Spanish is the most frequent language in the corpus, followed by English, Arabic, Portuguese, and the category undefined. Somewhat unexpected is the high proportion of undefined languages in the Twitter corpus, which has some major implications for our work. If the language detection algorithm used by Twitter fails to detect languages due to lack of sufficient text in the tweets, then it may be difficult for our MLLMs to detect violence from those signals. In contrast, if this behavior is due to the presence of unknown languages, then this is a good case scenario to showcase the benefits of using MLLMs.

```
[es, en, ar, pt, und, tl, ru, in, tr,  ja, id, fr, it, ht,
 ur, de, et, vi, nl,  sk, sl, pl, uk,  iw, fa, th, lv, ro,
 he, hu, fi, da, sv,  cy, lt, bg, no,  ko, hr, is, bs, ps,
 el, zh, hi, ka, sr,  ne, bn, iu, ckb, hy, ug, sn, pa, chr,
 te, sd, ta, am, bo,  kn, si, gu, lo,  or, ml, km]
```

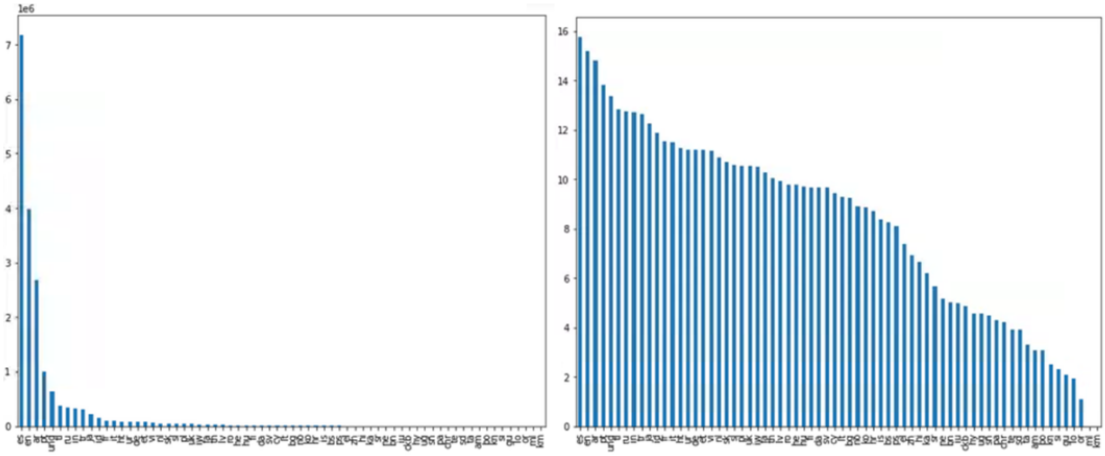

Figure 4.4. Language distribution in the corpus. (Top) List of ISO codes for the 68 languages in the Twitter corpus, including *und* for undefined. The labels are illustrated in the same order as they appear in the x-axis of the two bar plots. (Bottom-left) Language distribution in ordinal scale. (Bottom-right) Language distribution in logarithmnic scale.

4.2 Finding One: The Pre-violence Signal in a Tweet Is Stronger near the Location of a Future Violent Event

We began our work hypothesizing that a sufficient collective violence signal exists in the Twitter discourse prior to the occurrence of violent events. The closer the conversation

is to the violent events, in terms of time and distance, the stronger the presence of this signal. This hypothesis drove the process of constructing our training corpus for training the various models used in our project. We test our hypothesis using two test beds: projecting and visualizing the high-dimensional hidden states down to 2D, and plotting a labelwise ROC for an RF classifier.

The first method projects and then visualizes a sample of one million hidden state vectors from smaller-LaBSE into a lower-dimensional space. The choice of using smaller-LaBSE instead of XLM-T or LaBSE was completely arbitrary. When we pass the sample data through the network, a 768-dimensional hidden state vector is returned for each of the one million observations. Since visualizing any data in 768 dimensions is not intuitive, one way to escape this curse of dimensionality is to use a dimensionality reduction algorithm, such as the popular algorithm called Uniform Manifold Approximation and Projection (UMAP) (McInnes et al. 2018). With the help of UMAP, we project and visualize the 768-dimensional hidden state vector down to 2D. UMAP, like another popular algorithm called t-SNE, builds a neighbor graph in the original space of the data and tries to find a similar graph (projection) in lower dimensions. However, what sets UMAP apart from its competitors is the construction of the high-dimensional graph.

The graph density is estimated using KNN to each of the data points in the graph. Points with fewer neighbors in KNN are less densely connected than points with more neighbors (McInnes et al. 2018). Moreover, larger values of K better preserve global structure, while lower values better preserve local structures. There needs to be a balance, or trade-off, when choosing the optimal K, since this hyperparameter helps the algorithm estimate the graph density. In general, there is no formula to find the optimal value of K, some trial and error is required since K is dataset-dependent. Connections between each point and its neighbors are weighted with probabilities. Points further away from each other are weighted less than points which lie in close proximity. During the projection phase, the algorithm uses every point in high-dimensional space with its corresponding weighted edges, where high-weighted edges are more likely to stay together in the lower dimensional space (McInnes et al. 2018). For our project, we chose a value of K=2. UMAP is a computationally intensive algorithm which grows exponentially as the number of data points increases. Just to give an idea of this computational complexity, it took UMAP approximately four days and 10 hours to calculate the 2D projections for a 1M sample of the dataset.

Once the data is projected to a 2D space, we then plot the density of the two feature vectors—X and Y—across each binary category for the six labels individually, hoping for some separation in this lower-dimensional space, as illustrated in Figure 4.5. From this plot, we can barely see any clear patterns in our data. There is no clear separation both between the 0 and 1 categories within each label, and between the 0 or 1 categories across every other label. For example, there is no clear separation between categories 0 and 1 within *pre7geo10*, and similarly between the category 1 in *pre7geo30* and category 1 in *post7geo30*. Although we anticipated some patterns of separation, it is very hard to capture the entire underlying structure of a vector in 768 dimensions when reduced down to just two dimensions: "Just because some categories overlap does not mean that they are not separable in the original space. Conversely, if they are separable in the projected space they will be separable in the original space" (Tunstall et al. 2022, p. 43). Moreover, the model was not trained to contrast each label, "it only learned them implicitly by guessing the masked words in texts"(Tunstall et al. 2022, p. 43). This evidence suggests that we are facing a hard classification problem. We can only hope that any collective violence signal present in our data is separable in the high-dimensional space and that it will be strong enough to be captured by the multilabel classifiers. We will test this assertion next.

The second method to test our hypothesis is based on a labelwise ROC plot of an RF classifier trained on a 1.5M sample from the training set, as illustrated in Figure 4.6. Details of the classifier and its training procedure are addressed in the next subsection, but for now let us just focus on the plot. Note that the ROC-AUC scores are higher where the distance from the geolocated tweet is closer to the location of a violent event. For instance, *geo50* yields a smaller score (0.56) than the other two radii—*geo10* (0.61) and *geo30* (0.62)—for both *pre* and *post* violence labels. Surprisingly, this behavior is the opposite when we compare the two labels with the smallest radii, *geo10* and *geo30*. For example, one would expect *geo10* to perform better than *geo30* for being located much closer to ground zero, but in reality, *geo10* yields a slightly smaller score than *geo30*, 0.61 and 0.62, respectively. This observation also holds true for both *pre* and *post* violence labels.

A possible explanation for this last behavior might be attributed to the signal (collective violence) to noise ratio found in the geographical areas around each spatial distance from a tweet. The area surrounding *geo10* is approximately nine times smaller than the one for *geo30*. Therefore, it can be argued that the signal to noise ratio increases as the area

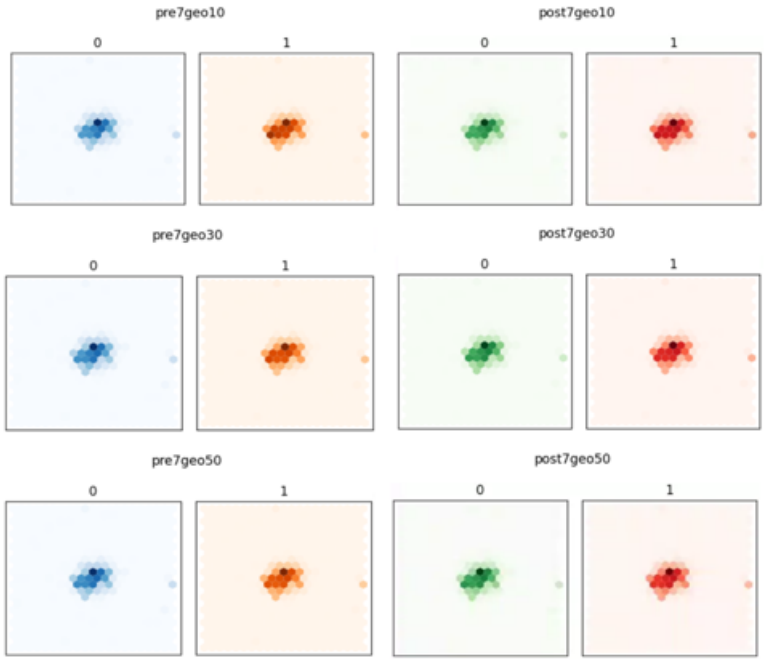

Figure 4.5. Visualization of the 768-d hidden state vectors from smaller-LaBSE in 2D, across each binary category for all six labels individually.

increases up to a point where the area is so large that this behavior reverses (i.e., *geo50*).[3] A careful consideration of which spatial heuristic is the most appropriate for detecting violence deserves a rigorous analysis, since we may end up training a classifier that either amplifies noise or the signal of interest.

The previous evidence provides support for our hypotheses in two ways. First, the observation that the classifier yields ROC-AUC scores well above 50%—or random guessing—means that the model is indeed picking up a collective violence signal in the Twitter discourse prior to the occurrence of violent events. There will be more to follow about other classifiers with better performances in the remainder of this section. Second, the fact that the ROC-AUC scores in Figure 4.6 generally decrease as the spatial distance increases suggests that the presence of the collective violence signal is stronger near the location of violent events. However, being too close to ground zero might slightly hurt the classifier performance due to noise hiding the signal in a relatively smaller amount of available data.

[3]Recall that the area of the circle is equal to πr^2.

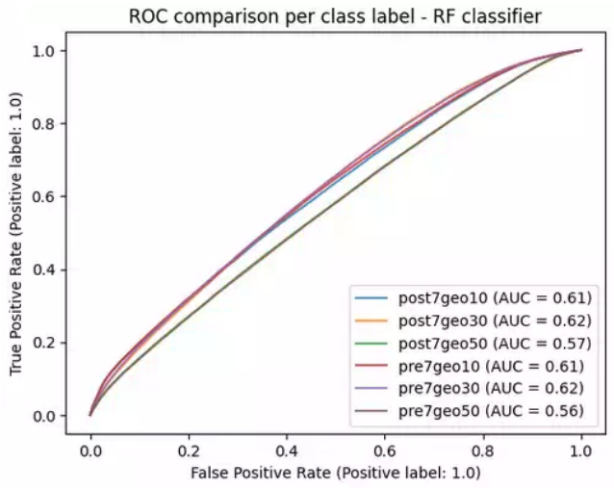

Figure 4.6. Labelwise ROC plot for an RF multilabel classifier. The *geo50* label, which corresponds to a tweet originating up to 50 kms away from the location of a violent event, obtains the lowest ROC-AUC score when compared to tweets originating in closer proximity.

4.3 Finding Two: Ensemble Multilabel Classifiers Are the Best Performing Traditional ML models

In this feature-based approach, we use the embeddings from three pretrained MLLMs (e.g., LaBSE, smaller-LaBSE, and XLM-T) to train five different traditional machine learning classifiers using scikit-learn (Pedregosa et al. 2011) and scikit-multilearn (Szymański and Kajdanowicz 2017) libraries. In short, we use the networks—or pretrained transformer models—as feature extractors, and pass these features to a regular classifier down the pipeline. Generally speaking, whenever we use a transformer model from the Hugging Face Hub for text classification as is, we use the network as end-to-end classifiers. Nonetheless, there is no rule that requires the sequence of text "to forward propagate through the *entire* network" (Rosebrock 2017a, p. 31). Instead, we can freeze the weights of any of the initial layers and stop propagation up to that point.

For our specific multilabel classification problem, we use the last hidden state of the pre-trained base models following recommended best practices in the NLP research community: "For classification tasks, it is common practice to just use the hidden state associated with the [CLS] token as the input feature" (Tunstall et al. 2022, p. 40). Moreover, we carefully chose five classifiers to account for the three most widely used techniques for multilabel

classification problems: problem adaptation, problem transformation, and ensemble methods. The idea is to evaluate which multilabel technique is best suited for the problem at hand (refer to Section 3.4.2 for a more thorough description of multilabel classifiers in machine learning). The classifiers evaluated in this section are as follows:

- *Problem Adaptation*: Bagging, Boosting, and support vector machine (SVM), using a Binary Relevance strategy in all of them.
- *Problem Transformation*: DT and RF.
- *Ensemble Models*: Three of the five classifiers also belong to this category (e.g., Bagging, Boosting, and RF); however, their training strategies differ from one another and normally fall under one of the two techniques described above.

Table 4.1 summarizes the results after training the previous five multilabel classifiers on the dataset. The inputs to the classifiers are the last hidden states from the transformer models, which were obtained after tokenizing a random sample of 1.5 million instances from the training split. All classifiers were trained using the default parameters in their corresponding implementations in scikit-learn and scikit-multilearn. The main insights from Table 4.1 show that, in general, the best performing classifiers across all seven different metrics are *ensemble models* (for a more detailed explanation of how multilabel metrics are implemented and their correct interpretation, refer to Section 3.5). Moreover, *ensemble models* trained using a *problem transformation* approach not only perform better than their *problem adaptation* counterparts, but they are also computationally faster. For example, RF is the best performing model in terms of ROC-AUC, whose performance (0.6028) is slightly better than the second best model *bagging* (0.5938); however, the latter algorithm takes approximately 5 times longer to train.

Just to give an idea of the computational complexity, all five models, except DT, were parallelized during training using all CPU cores. Surprisingly, even though the scikit-learn implementation of DT is not parallelizable, it is the fastest algorithm of all. Moreover, features extracted from XLM-T yield better performances than LaBSE and smaller-LaBSE across all five classifiers and all seven metrics, except for two cases: the weighted-precision metric for the *boosting* classifier, and precision-micro for SVM. For what follows in the remainder of this chapter, we will only use ROC-AUC scores to compare different classifiers. For this particular test bed, RF trained on XLM-T features is the best performing model,

whose ROC-AUC score (0.6028) is slightly better than LaBSE (0.5903) and smaller-LaBSE (0.5887). This traditional machine learning model will serve as a baseline for comparing the performance of state-of-the art deep learning models next.

Table 4.1. Summary of results of five classical ML multilabel classifiers. RF performs better than any other model across all seven metrics. In terms of computational time, RF is the second fastest model next to Decision Trees.

Metric	Tokenizer	Bagging	Boosting	SVM-BR	D.T.	R.F.
ROC-AUC	s-LaBSE	0.5564	0.5845	0.5168	0.5393	0.5902
	LaBSE	0.5555	0.5845	0.5096	0.5391	0.5903
	XLM-T	0.5682	0.5938	0.5841	0.5474	**0.6028**
Precision (weighted)	s-LaBSE	0.5186	0.6118	0.2906	0.4992	0.5006
	LaBSE	0.5181	0.6112	0.2695	0.4988	0.5712
	XLM-T	0.5382	0.5858	0.6012	0.5079	0.5824
Recall (weighted)	s-LaBSE	0.4094	0.4712	0.0769	0.4991	0.4923
	LaBSE	0.4066	0.4736	0.0404	0.5004	0.4922
	XLM-T	0.4324	0.52	0.4763	0.5104	0.5322
F1 (weighted)	s-LaBSE	0.4475	0.3903	0.0953	0.4990	0.4515
	LaBSE	0.4454	0.3917	0.0616	0.4994	0.4511
	XLM-T	0.4704	0.4532	0.411	0.509	0.4983
Precision (micro)	s-LaBSE	0.5389	0.57	0.6029	0.4997	0.5726
	LaBSE	0.5388	0.5691	0.6184	0.5017	0.573
	XLM-T	0.5538	0.5728	0.5677	0.5106	0.5806
Recall (micro)	s-LaBSE	0.1065	0.4708	0.0774	0.4978	0.4912
	LaBSE	0.4066	0.4736	0.0404	0.5004	0.4922
	XLM-T	0.4324	0.52	0.4763	0.2104	0.5322
F1 (micro)	s-LaBSE	0.4634	0.5157	0.1372	0.4988	0.5288
	LaBSE	0.4635	0.517	0.0758	0.501	0.5295
	XLM-T	0.4856	0.5451	0.518	0.5105	0.5553
Training time	s-LaBSE	5h 6m 55s	20h 8m 16s	5h 53m 40s	1h 33m 4s	4h 17m 22s
	LaBSE	5h 1m 13s	22h 12m 50s	5h 41m 46s	1h 29m 33s	4h 13m 47s
	XLM-T	6h 12m 7s	21h 32m 4s	8h 17m 34s	1h 53m 21s	4h 30m 5s

4.4 Finding Three: XLM-T, LaBSE, and Smaller-LaBSE Perform Similarly

In this section, we will explore the performances of deep learning models using a fine-tuning approach, where we train three end-to-end MLLMs on our full training corpus: LaBSE,

smaller-LaBSE, and XLM-T. Fine tuning, in contrast to feature extraction, propagates the errors back to the network, calculates gradients, and updates the weight parameters of the feed-forward classifier (Tunstall et al. 2022, p. 38). The training process starts by building a multilabel classification head on top of the base model architecture. We refer the reader to Section 3.4 for a thorough explanation of transformer architectures as well as fine-tuning transformer models. In the world of data science, the Hugging Face Hub is the starting point in the NLP domain, offering its library of popular transformer models created by companies like Apple, Microsoft, and Google, with seamless implementations. Fortunately, all three models used in our work are available at the Hub, which means that we do not need to code the networks ourselves. Moreover, the classification head *class* supports multilabel classification natively. All models were trained using the parameters detailed as follows:

- Model checkpoints from the Hugging Face Hub: setu4993/LaBSE, setu4993/smaller-LaBSE, and cardiffnlp/twitter-xlm-roberta-base
- Head class: AutoModelForSequenceClassification
- Training batch size = 1024
- Validation batch size = 1024
- Number of epochs = 20
- Learning rate scheduler type = cosine
- Weight decay = 0.1
- Initial learning rate = 5e-5
- Random seed = 42
- Max. sequence length = 32
- Saving strategy = save the best model only when there is improvement on the ROC-AUC score on the validation split

Table 4.2 summarizes the results after training the three selected MLLMs on the entire dataset. All models were evaluated on the full validation split using the ROC-AUC score. These results show that XLM-T takes approximately 45 minutes and 90 minutes longer to train when compared to smaller-LaBSE and LaBSE, respectively. This difference in training time is somewhat expected, since XLM-T has approximately 100 million more parameters than LaBSE (see Table 2.1 for more details about the different features of each MLLM). Moreover, the three models yield similar performances in terms of ROC-AUC score, each scoring approximately 0.73.

Table 4.2. Performance results after fine tuning three multilingual models: LaBSE, smaller-LaBSE, and XLM-T.

Model	lr	roc_auc	training time
smaller-LaBSE	0.00004969	0.7246	7h 31m 54s
LaBSE	0.00004983	0.7238	8h 3m 48s
XLM-T	0.00004973	0.7268	8h 48m 14s

Our initial suspicion was that XLM-T would outperform the other two models, because its vocabulary corresponds to the informal language found in a Twitter conversation. Recall that XLM-T was pre-trained exclusively on Twitter data, in stark contrast to LaBSE which was trained on Wikipedia and CC. One would expect to find a vocabulary full of emojis in XLM-T and a lack thereof in LaBSE. A visualization of a sample of single-character tokens in both vocabularies may shed some light on this, as illustrated in Figure 4.7. It follows from this figure that both LaBSE and XLM-T share a similar number of single-character tokens, approximately 14,000, and a good portion of them correspond to emojis. This evidence suggests that the presence of emojis and the like in the vocabulary of LaBSE is what allows this model to perform well even on data very different from what it was trained on.

Figure 4.7. Comparison of single-character tokens in the vocabularies of LaBSE and XLM-T. Both MLLMs share a similar number of single-character tokens and a good portion of them correspond to emojis.

By carefully inspecting Figures 4.8, 4.9, and 4.10—smaller-LaBSE, LaBSE, and XLM-T respectively—we note that the final models do not show any signs of overfitting across the 20 epochs. The training and validation losses are constantly decreasing over time. Likewise, there is no pronounced departure of the validation loss from the training loss. Moreover, the ROC-AUC score starts to stagnate at approximately epoch 15. It could be argued that all these positive benefits are in part due to the use of a learning rate scheduler with a weight decay during training: "By adjusting our learning rate on an epoch-to-epoch basis, we can reduce loss, increase accuracy, and even in certain situations reduce the total amount of time it takes to train a network" (Rosebrock 2017b, p. 241). This evidence suggests that the model would not learn past 20 epochs without the risk of overfitting the training data.

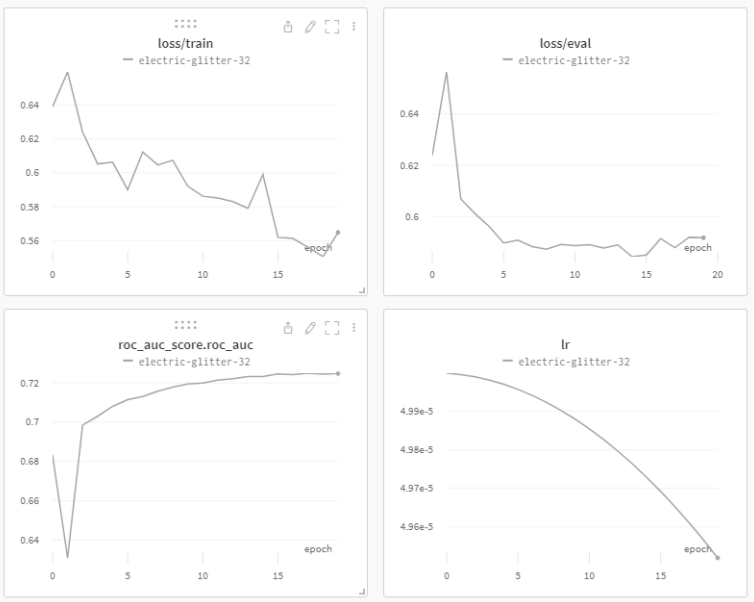

Figure 4.8. Performance results of smaller-LaBSE. All metrics are evaluated across 20 epochs. (Top-left) Train loss. (Top-right) Evaluation loss. (Bottom-left) ROC-AUC score. (Bottom-right) Learning rate decay.

If we were to pick the best performing MLLM for violence prediction, it is obvious that the ROC-AUC score would not provide sufficient information as a criterion for selecting a good candidate, as it is nearly equal across the candidates. It then follows that we need another selection criterion, perhaps one related to the memory footprint of each model in GPU. Let us keep in mind that all models were trained in a server with 16 Tesla V-100 GPUs with 32 GB of memory each. Figure 4.11 shows a comparison of the GPU memory

Figure 4.9. Performance results of LaBSE. All metrics are evaluated across 20 epochs. (Top-left) Train loss. (Top-right) Evaluation loss. (Bottom-left) ROC-AUC score. (Bottom-right) Learning rate decay.

Figure 4.10. Performance results of XLM-T. All metrics are evaluated across 20 epochs. (Top-left) Train loss. (Top-right) Evaluation loss. (Bottom-left) ROC-AUC score. (Bottom-right) Learning rate decay.

allocation for each MLLM, amounting to 60%, 70%, and 80% for smaller-LaBSE, XLM-T, and LaBSE, respectively. Likewise, the spikes of GPU percent utilization rarely cross the 85% and 95% thresholds for smaller-LaBSE and XLM-T, respectively; whereas there is an instance of LaBSE that approaches almost 100%.

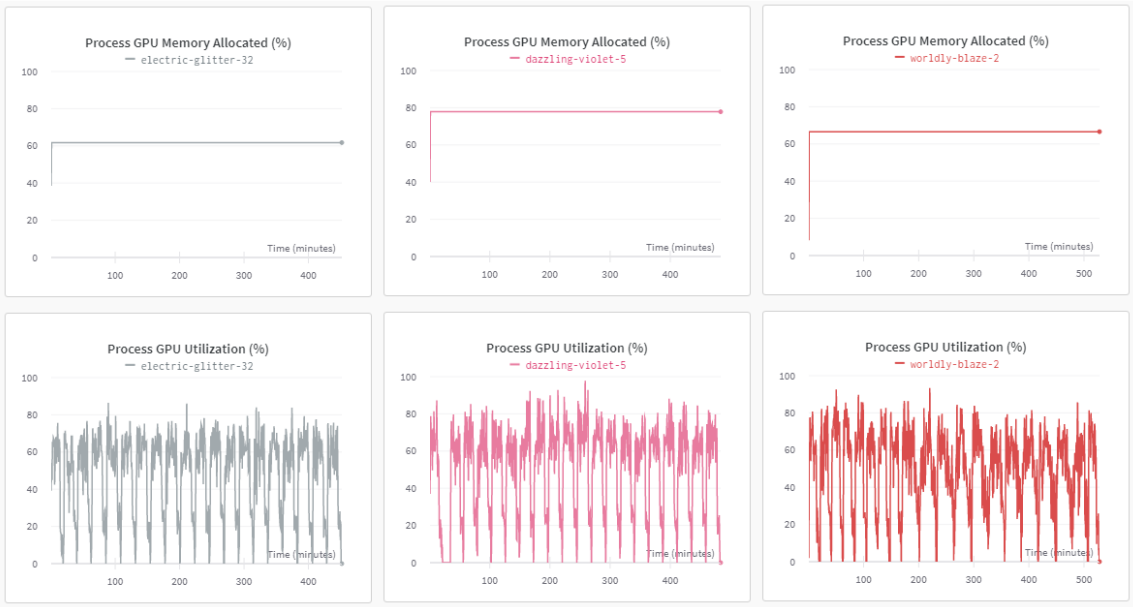

Figure 4.11. GPU performance metrics when training three multilingual models. (Left) smaller-LaBSE. (Center) LaBSE. (Right) XLM-T.

These results are expected because LaBSE, despite being a smaller model than XLM-T in terms of parameters, has a vocabulary twice as large; therefore, it requires more memory and computing resources. To illustrate the importance of model sizes in any NLP application, training LaBSE and smaller-LaBSE in a Tesla V-100 GPU with maximum sequence length of 32 and batch sizes greater than 1024 yields an *out of memory* error. This could be a potential problem during inference time when making final predictions. It follows from this evidence that the best MLLM for violence prediction, in terms of memory footprint and performance, is smaller-LaBSE. On a final note before proceeding to the next section, it is worth mentioning that all three model checkpoints, as well as the dataset, are publicly available in the Hugging Face Hub at *https://huggingface.co/m2im*.

4.5 Finding Four: Deep Learning MLLMs Outperform Traditional ML Models

This finding follows from the two previous findings. It does not take a great deal of analysis to compare results from Tables 4.1 and 4.2 before realizing that deep learning MLLMs outperform traditional ML models by approximately 13% in terms of ROC-AUC score. Nonetheless, jumping abruptly to this conclusion before crafting more rigorous experimentation would do traditional ML models no justice. Let us recall that all traditional ML classifiers were trained with their default parameter configurations. A more fair comparison would require some sort of hyperparameter optimization of the best performing traditional ML algorithm; this is RF trained on XLM-T features. Following are the multiple configuration parameters used to perform random search hyperparameter tuning on RF:

- Metric: ROC-AUC score
- Goal: maximize
- Method: random search
- Number of combinations: 20
- Criterion: gini, entropy
- Num_estimators: [50, 500, 1000]
- Min_samples_split: [10, 100, 1000]
- Min_samples_leaf: [10, 50, 100]

The left picture in Figure 4.12 shows a plot of the different ROC-AUC scores obtained during the random search. These results show that of all 20 different combinations, only four models yield the highest performance (0.6022), whose configurations are detailed in Table 4.3. Note that every other model in Table 4.3 has the same configuration except for the *criterion* parameter. For example, model 1 uses *entropy* as opposed to model 3 that uses *gini*, with the remaining parameters being the same. This is also the case for model 2 and model 4. This behavior is explained with the feature importance and correlation plot illustrated in the right picture of Figure 4.12. It appears that the most important parameter for this RF model is *min_samples_split*, whereas *criterion* plays almost no role. Figure 4.13 provides a full visualization of how each model in the random search performs.

Our work started by asking the research question: "To what extent can the information contained in social media (Twitter) generate sufficient signals for use in multilingual trans-

Figure 4.12. Results of a random search for hyperparameter tuning for the RF model trained on XLM-T features. (Left) Plot of the ROC-AUC score for the 20 different combinations. (Right) Feature importance and Correlation for the hyperparameters tuned in the model.

Table 4.3. Configuration parameters of the best performing RF models after random search. All four models yield an ROC-AUC score of 0.6022.

Parameters	**Model 1**	**Model 2**	**Model 3**	**Model 4**
Criterion	entropy	entropy	gini	gini
min_samples_leaf	10	10	10	10
min_samples_split	100	10	100	10
num_estimators	1000	50	1000	50

former models to predict collective violence events efficiently?" The results presented here show that regardless of the hyperparameter configurations of RF, none of the best performing traditional machine learning models obtained thus far seem to improve over the baseline performance achieved by an RF trained with default configurations (0.6028); in fact, their performances are similar (see Table 4.1). Simply put, it appears that RF has reached its performance limit no matter how much we tweak its hyperparameters. This evidence demonstrates that traditional ML models do not perform as well as deep learning MLLMs for predicting the outcome of collective violence. Moreover, this finding also illustrates that MLLMs are well suited to capture the sparse collective violence signals contained in social media discourse, thus answering the research question that drove our research endeavor.

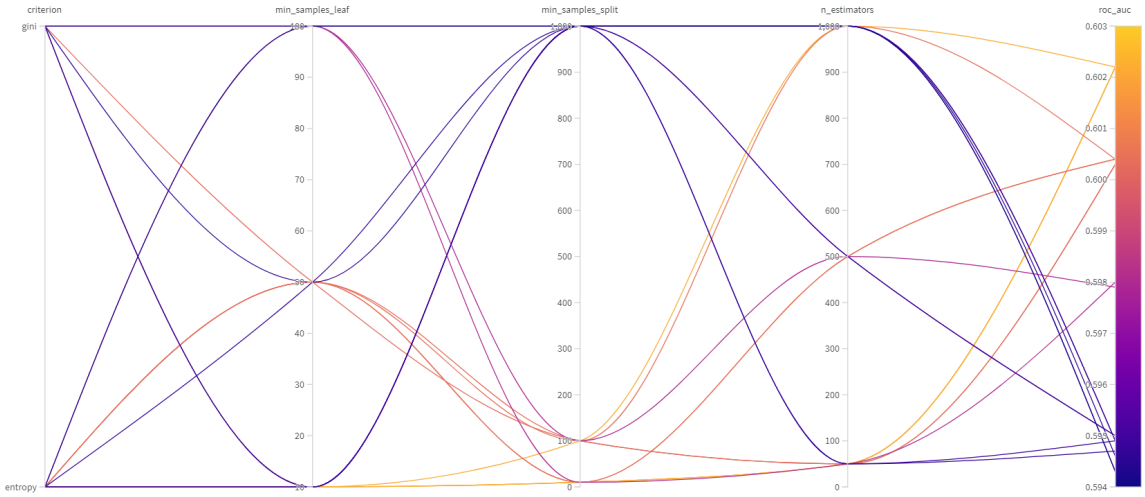

Figure 4.13. Parallel coordinate plot with the hyperparameter optimization values for the RF model. Each line corresponds to a different combination of hyperparameters. Model performance is measured in terms of ROC-AUC score.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusions and Further Avenues of Research

5.1 Conclusions

In this thesis, we investigated the utility of conducting social media analysis through the development of language-agnostic deep neural networks. The reason for such a model is twofold. First, social media users come widely diverse linguistic communities. Second, in social media discourse people often alternate between different languages within text even within the same conversation. By constructing a model that can accept input from social media messages written in any language and learn to associate forms of discourse with the timing and location of events of collective violence, we generated vector embeddings that capture semantic elements more likely to occur in the periods immediately preceding and immediately following violent events. Our aim wass that this will provide a useful representation of violent discourse, which might prove beneficial for predicting violence through social media.

One of the major challenges in our research was the lack of a suitable social media dataset for the classification task. We built our corpus under the hypothesis that sufficient collective violence signals exist in the Twitter discourse prior to the occurrence of violent events. The closer the conversation is to violent events in terms of time and distance, the stronger the presence of this signal. We limited our analysis to the one-year period from August 1, 2013, to July 31, 2014, to match the timespan of a historical archive of Twitter messages licensed for research at NPS. We gathered a dataset of approximately 23 million tweets around a curated list of 10,071 unique violent events worldwide from the Uppsala Conflict Data Program (UCDP) dataset (Sundberg and Melander 2013). This wide variety of locations implied a broad linguistic diversity, which motivated the use of MLLMs in our research. To account for the sparsity of the violence signal contained in the training data when compared to the signal of non-violent events, we framed the problem as a multilabel classification problem, with the hope that the model would pick up and amplify violence signals across different spatial-temporal scales.

Our work started by asking the research question, "To what extent can the information contained in social media (Twitter) generate sufficient signals to efficiently predict collective violence events using multilingual transformer models?" We postulated this question because there is a gap in understanding collective violence in multiple or mixed languages in social media. We answered this question by comparing the performance of traditional ML classifiers against deep learning MLLMs. In both cases, we used three MLLMs (i.e., smaller-LaBSE, LaBSE, and XLM-T) to either extract features from our data or to serve as end-to-end classifiers. In the former case, we chose five ML classifiers that represent the three most widely used techniques for multilabel classification: problem adaptation, problem transformation, and ensemble models. Likewise, for the deep learning approach, we fined tuned smaller-LaBSE, LaBSE, and XLM-T using pretrained checkpoints available in the Hugging Face Hub.

The results of the models produced four noteworthy findings. *Finding one* confirmed our hypothesis that the pre-violence signal in a tweet is stronger near the location of a future violent event. A labelwise ROC plot of an RF classifier trained on a 1.5M sample shows that ROC-AUC scores decrease as the spatial distance increases, suggesting that the presence of the collective violence signal is stronger near the location of violent events. However, the same plot also indicated that being too close to ground zero might hurt the classifier performance slightly due to noise hiding the signal in a relatively smaller geographical area. *Finding two* revealed that ensemble multilabel classifiers are the best performing traditional ML models across the seven different metrics used in our benchmark. These models were trained using the last hidden state of the pretrained MLLMs as input features to the classifiers. Interestingly, ensemble models trained using a problem transformation approach not only performed better than their problem adaptation counterparts, but they were also computationally faster. Furthermore, features extracted from XLM-T yielded better performances than LaBSE and small-LaBSE across all five classifiers and all seven metrics. In the end, RF trained on XLM-T features was the best performing traditional ML model with a ROC-AUC score of 0.6028.

The *third finding* indicated that smaller-LaBSE, LaBSE, and XLM-T yielded similar performances in terms of ROC-AUC scores, each scoring approximately 0.73. In terms of memory footprint, smaller-LaBSE was the best performing model due to a smaller vocabulary size and less number of parameters than the other two MLLMs. The *last finding* followed from

the previous two findings: deep learning MLLMs outperformed traditional ML models by approximately 13% in terms of ROC-AUC scores. This drastic skill difference was maintained even after performing a random search hyperparameter optimization on the best traditional ML classifier. It appears as if RF reached its maximum performance limit no matter how much we tuned its hyperparameters. This evidence indicates that traditional ML models do not perform as well as deep learning MLLMs for violence prediction. Moreover, it follows from the last finding that MLLMs are well suited to capture the sparse collective violence signals present in a Twitter discourse, thus answering the research question that drove our research effort.

Ultimately, our findings also suggest that a predictive relationship between social media content and past events of collective violence exists. Moreover, the relationship between social media content and collective violence can be observed regardless of the language used in the discourse. We believe that our study could serve as a useful decision-aid tool in the military for predicting violence on a global scale at the operational and strategic levels of war. Samuel Huntington, in his famous book titled *The Soldier and the State: The Theory and Politics of Civil-Military Relations*, drawing on a quotation from Harold Laswell, argued that the military expertise of the officer corps revolves around the "management of violence" (1981, p. 21). With more than 7,000 languages spoken in the world today, and an ever-increasing responsibility on the officer corps to cope with violence everywhere, one major contribution of our work is that military commands now have a tool to evaluate and learn the language of violence across all human languages, thus reducing the negative effects of hostile information campaigns in social media (David et al. 2022). In future work, we recommend re-training the previously developed *U-Net* conflict prediction model, taking as new inputs the output of the fine-tuned XLM-T model, and comparing performance differences. Finally, we made the code, data, and models publicly available at *https://huggingface.co/m2im/*, with the hope that this will help the research community advance its efforts in conflict prediction in addition to enabling our warfighters to use the model as a tool to enhance their understanding of the information environment.

5.2 Further Avenues of Research

Our work identified six avenues of potential future research. First, we recommend replicating the same experiments using a different set of spatial-temporal heuristics to determine which set of labels better captures a collective violence signal prior to the occurrence of a violent event. Let us recall that we collected 40 different combinations of spatial-temporal dependencies and used only six of those combinations as the target labels in our classification task. Our research only addressed one temporal heuristic—seven days—combined with three different spatial distances—10, 30, and 50 kms—for both pre- and post-violence events. Understanding which spatial-temporal heuristics provide a better predictive relationship could illuminate how violence-related events unfold on the ground. Second, incorporating sentiment from tweets might help filtering unwanted noise signals. This recommendation is based on the findings of two previous NPS theses, in which the authors independently demonstrated the existence of a relationship between negative sentiments and the future outcomes of violence in two different scenarios, Iraq and Ukraine, respectively (Frost et al. 2017; Kuah and Chew 2018). We recommend further experiments using two classifiers in cascade, one for sentiment and the other for multilabel predictions. Under this setting, only tweets with negative sentiments would feed into the second classifier for behavioral predictions of collective violence.

Third, incorporating extreme gradient boosting (XGBoost) into the set of traditional ML classifiers and then comparing results with those of the MLLMs might make a fairer comparison. Unlike the other traditional classifiers used in our benchmark, XGBoost is GPU-enabled. This nice feature together with the fact that its memory footprint is orders of magnitude smaller than any existing MLLM might prove beneficial during inference time. We recommend crafting additional experimentation to measure the computational performance differences between XGBoost and the other deep learning models. Fourth, performing hyperparameter optimization for the three fine-tuned MLLMs might help in determining which model is better equipped to predict violence from Twitter conversations. Perhaps by changing some hyperparameters (e.g., learning rate, weight decay, number of epochs) we might appreciate some further deviations in performance from one model to another. We hypothesize that XLM-T should yield the best performance since it was pretrained on Twitter data, but this claim requires additional experimentation.

Fifth, we recommend training from scratch either LaBSE, smaller-LaBSE, or any other MLLM on our Twitter corpus and then comparing results with those obtained by XLM-T to make a fairer comparison. This is a very expensive operation since it requires training a tokenizer, pre-training a masked language model, and finally fine-tuning the models as multilabel classifiers. The fact that the only MLLM trained on Twitter data is XLM-T makes this recommendation very appealing. And lastly, future work should validate the results obtained by the best performing MLLM using the *U-Net* architecture developed by the CODA lab. Let us recall that an important goal of this thesis is to find a model that would improve the predictive power of the Twitter counts fed into *U-Net* by using MLLMs. Any performance boost obtained through this approach could prove beneficial for the core task of predicting collective violence on a global scale.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Abdaoui A, Pradel C, Sigel G (2020) Load what you need: Smaller versions of multilingual BERT. arXiv preprint arXiv:2010.05609 https://doi.org/10.48550/arXiv.2010.05609.

Abid A, Tunstall L, Carrigan M, Debut L, Gugger S, Khan D, Noyan M, Saulnier L (2022) The Hugging Face course. Hugging Face. Accessed July 10, 2022, https://huggingface.co/course/.

Alizadeh M, Shapiro JN, Buntain C, Tucker JA (2020) Content-based features predict social media influence operations. *Science advances* 6(30):eabb5824.

Antypas D, Preece A, Camacho-Collados J (2022) Politics, sentiment and virality: A large-scale multilingual Twitter analysis in Greece, Spain and United Kingdom. *SSRN* http://dx.doi.org/10.2139/ssrn.4166108.

Arghyadeep D (2021) Multi-label emotion classification with pytorch + Huggingface's transformers and W&B for tracking. Towards Data Science. Accessed September 1, 2022, https://towardsdatascience.com/multi-label-emotion-classification-with-pytorch-huggingfaces-transformers-and-w-b-for-tracking-a060d817923.

Barbieri F, Camacho-Collados J, Espinosa-Anke L, Neves L (2020) Tweeteval: Unified benchmark and comparative evaluation for tweet classification. arXiv abs/2010.12421, https://doi.org/10.48550/arXiv.2010.12421.

Barbieri F, Espinosa-Anke L, Camacho-Collados J (2022) XLM-T: Multilingual language models in twitter for sentiment analysis and beyond. *Proceedings of the LREC, Marseille, France* 20–25.

Beltagy I, Lo K, Cohan A (2019) Scibert: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676 https://doi.org/10.48550/arXiv.1903.10676.

Brandt PT, D'Orazio V, Khan L, Li YF, Osorio J, Sianan M (2022) Conflict forecasting with event data and spatio-temporal graph convolutional networks. *International Interactions* 48(4):1–23, https://doi.org/10.1080/03050629.2022.2036987.

Breiman L (1996) Bagging predictors. *Machine Learning* 24(2):123–140.

Breiman L (2001) Random forests. *Machine Learning* 45(1):5–32.

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. *International Group* 432(151-166):9.

Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al. (2020) Language models are few-shot learners. *Advances in neural information processing systems* 33:1877–1901.

Cahyawijaya S, Winata GI, Wilie B, Vincentio K, Li X, Kuncoro A, Ruder S, Lim ZY, Bahar S, Khodra ML, et al. (2021) Indonlg: Benchmark and resources for evaluating indonesian natural language generation. arXiv preprint arXiv:2104.08200 https://doi.org/10.48550/arXiv.2104.08200.

Celiku B, Kraay A (2017) Predicting conflict. *World Bank Policy Research Working Paper* (8075), https://ssrn.com/abstract=2985500.

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 785–794 (San Francisco, CA), https://doi.org/10.1145/2939672.2939785.

Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Grave E, Ott M, Zettlemoyer L, Stoyanov V (2019) Unsupervised cross-lingual representation learn-ing at scale. arXiv preprint arXiv:1911.02116 https://doi.org/10.48550/arXiv.1911. 02116.

David E, Simons G, Fennig C (2022) Ethnologue: Languages of the world, 25th edition. SIL International. Accessed October 3, 2022, https://www.ethnologue.com/guides/how-many-languages.

Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv abs/1810.04805, https://doi.org/10.48550/arXiv.1810.04805.

Doddapaneni S, Ramesh G, Kunchukuttan A, Kumar P, Khapra MM (2021) A primer on pretrained multilingual language models. arXiv abs/2107.00676, https://doi.org/10.48550/arXiv.2107.00676.

Everton S (2012) *Disrupting dark networks*. Number 34 (Cambridge University Press, United Kingdom).

Fedus W, Zoph B, Shazeer N (2022) Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research* 23(120):1–39, http://jmlr.org/papers/v23/21-0998.html.

Feng F, Yang Y, Cer D, Arivazhagan N, Wang W (2020) Language-agnostic BERT sentence embedding. arXiv https://doi.org/10.48550/arXiv.2007.01852.

Friedman JH (2001) Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 1189–1232.

Frost H, Evans A, Hodges R (2017) *Understanding violence through social media*. Master's thesis, Department of Defense Analysis, Naval Postgraduate School, Monterey, CA, http://hdl.handle.net/10945/56920.

Gencoglu O (2020) Large-scale, language-agnostic discourse classification of tweets during covid-19. *Machine Learning and Knowledge Extraction* 2(4):603–616.

Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. *Machine learning* 63(1):3–42.

Goyal N, Du J, Ott M, Anantharaman G, Conneau A (2021) Larger-scale transformers for multilingual masked language modeling. arXiv preprint arXiv:2105.00572 https: //doi.org/10.48550/arXiv.2105.00572.

Hastie T, Tibshirani R, Friedman JH, Friedman JH (2009) *The elements of statistical learning: data mining, inference, and prediction* (Springer).

Hegre H, Allansson M, Basedau M, Colaresi M, Croicu M, Fjelde H, Hoyles F, Hultman L, Högbladh S, Jansen R, et al. (2019) Views: A political violence early-warning system. *Journal of Peace Research* 56(2):155–174.

Hegre H, Bell C, Colaresi M, Croicu M, Hoyles F, Jansen R, Leis MR, Lindqvist-McGowan A, Randahl D, Rød EG, et al. (2021) Views2020: revising and evaluating the views political violence early-warning system. *Journal of Peace Research* 58(3):599–611.

Hegre H, Karlsen J, Nygård HM, Strand H, Urdal H (2013) Predicting armed conflict, 2010–2050. *International Studies Quarterly* 57(2):250–270.

Högbladh S (2022) UCDP GED codebook version 22.1. Department of Peace and Conflict Research, Uppsala University. Accessed June 20, 2022, https://ucdp.uu.se/downloads/ged/ged221.pdf.

Huang H, Liang Y, Duan N, Gong M, Shou L, Jiang D, Zhou M (2019) Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. arXiv preprint arXiv:1909.00964 https://doi.org/10.48550/arXiv.1909.00964.

Huntington S (1981) *The soldier and the state: The theory and politics of civil-military relations* (Harvard University Press, Cambridge, MA).

Islam MR, Liu S, Wang X, Xu G (2020) Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining* 10(1):1–20.

Joint Chiefs of Staff (2006) Information operations. JP 3-13, Washington, DC, https://www.globalsecurity.org/intell/library/policy/dod/joint/jp3_13_2006.pdf.

Kalyan KS, Rajasekharan A, Sangeetha S (2021) Ammus: A survey of transformer-based pretrained models in natural language processing. arXiv abs/2108.05542, https://doi.org/10.48550/arXiv.2108.05542.

Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, Gray S, Radford A, Wu J, Amodei D (2020) Scaling laws for neural language models. arXiv abs/2001.08361, https://doi.org/10.48550/arXiv.2001.08361.

Karakaya M (2020) Multi label model evaluation. Kaggle. Accessed September 10, 2022, https://www.kaggle.com/code/kmkarakaya/multi-label-model-evaluation.

Khanuja S, Bansal D, Mehtani S, Khosla S, Dey A, Gopalan B, Margam DK, Aggarwal P, Nagipogu RT, Dave S, et al. (2021) Muril: Multilingual representations for Indian languages. arXiv abs/2103.10730, https://doi.org/10.48550/arXiv.2103.10730.

Kuah W, Chew YHW (2018) *Hashtags warriors: the influence of social social media on collective violence in Ukraine*. Master's thesis, Department of Defense Analysis, Naval Postgraduate School, Monterey, CA, http://hdl.handle.net/10945/61332.

Kudo T, Richardson J (2018) Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226 https://doi.org/10.48550/arXiv.1808.06226.

Lample G, Conneau A (2019) Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291* https://doi.org/10.48550/arXiv.1901.07291.

Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2019) Albert: A lite BERT for self-supervised learning of language representations. arXiv abs/1909.11942, https: //doi.org/10.48550/arXiv.1909.11942.

Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J (2020) Biobert: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36(4):1234–1240.

Lepikhin D, Lee H, Xu Y, Chen D, Firat O, Huang Y, Krikun M, Shazeer N, Chen Z (2020) Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv abs/2006.16668, https://doi.org/10.48550/arXiv.2006.16668.

Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2019) BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 https://doi.org/10.48550/arXiv.1910.13461.

Lin X, Lin N, Wattanachote K, Jiang S, Wang L (2021) Multilingual text classification for Dravidian languages. arXiv abs/2112.01705, https://doi.org/10.48550/arXiv.2112.01705.

Liu Y, Gu J, Goyal N, Li X, Edunov S, Ghazvininejad M, Lewis M, Zettlemoyer L (2020a) Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics* 8:726–742.

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized BERT pretraining approach. arXiv abs/1907.11692, https://doi.org/10.48550/arXiv.1907.11692.

Liu Z, Winata GI, Madotto A, Fung P (2020b) Exploring fine-tuning techniques for pretrained cross-lingual models via continual learning. arXiv abs/2004.14218, https://doi.org/10.48550/arXiv.2004.14218.

Mazarr MJ, Casey A, Demus A, Harold SW, Matthews LJ, Beauchamp-Mustafaga N, Sladden J (2019) Hostile social manipulation present realities and emerging trends https://apps.dtic.mil/sti/citations/AD1081269.

McInnes L, Healy J, Melville J (2018) UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv abs/1802.03426, https://doi.org/10.48550/arXiv.1802.03426.

Mitts T, Phillips G, Walter BF (2022) Studying the impact of ISIS propaganda campaigns. *The Journal of Politics* 84(2):1220–1225.

Mooijman M, Hoover J, Lin Y, Ji H, Dehghani M (2018) Moralization in social networks and the emergence of violence during protests. *Nature human behaviour* 2(6):389–396.

Müller K, Schwarz C (2021) Fanning the flames of hate: Social media and hate crime. *Journal of the European Economic Association* 19(4):2131–2167.

Müller M, Salathé M, Kummervold PE (2020) Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. arXiv abs/2005.07503, https://doi.org/10.48550/arXiv.2005.07503.

Nguyen DQ, Vu T, Nguyen AT (2020) BERTweet: A pre-trained language model for English tweets. *arXiv* abs/2005.10200, https://doi.org/10.48550/arXiv.2005.10200.

Pacheco D, Hui PM, Torres-Lugo C, Truong BT, Flammini A, Menczer F (2021) Uncovering coordinated networks on social media: Methods and case studies. *ICWSM* 21:455–466.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Pei Y, Chen S, Ke Z, Silamu W, Guo Q (2022) Ab-LaBSE: Uyghur sentiment analysis via the pre-training model with bilstm. *Applied Sciences* 12(3):1182.

Popescu N, Secrieru S (2018) *Hacks, Leaks and Disruptions*. https://www.jstor.org/stable/pdf/resrep21140.1.pdf.

Python A, Bender A, Nandi AK, Hancock PA, Arambepola R, Brandsch J, Lucas TC (2021) Predicting non-state terrorism worldwide. *Science Advances* 7(31):eabg4778.

Qiao L, Xiangsui W, Wang X (2002) *Unrestricted warfare: China's master plan to destroy America* (NewsMax Media, Inc., Boca Raton, FL).

Radford A, Narasimhan K, Salimans T, Sutskever I, et al. (2018) Improving language understanding by generative pre-training https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf.

Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al. (2019) Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ, et al. (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21(140):1–67.

Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. *Machine Learning* 85(3):333–359.

Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-assisted Intervention*, 234–241 (Springer).

Rosebrock A (2017a) *Deep Learning for Computer Vision with Python: Practitioner Bundle*. Deep learning for computer vision with Python (PyImageSearch, MD).

Rosebrock A (2017b) *Deep Learning for Computer Vision with Python: Starter Bundle*. Deep learning for computer vision with Python (PyImageSearch, MD).

Sennrich R, Haddow B, Birch A (2015) Neural machine translation of rare words with subword units. arXiv abs/1508.07909, https://doi.org/10.48550/arXiv.1508.07909.

Siddhant A, Bapna A, Firat O, Cao Y, Chen MX, Caswell I, Garcia X (2022) Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. arXiv abs/2201.03110, https://doi.org/10.48550/arXiv.2201.03110.

Sorower MS (2010) A literature survey on algorithms for multi-label learning. Oregon State University, Corvallis 18(1):25.

Souza F, Nogueira R, Lotufo R (2020) BERTimbau: Pretrained BERT models for Brazilian Portuguese. *Brazilian Conference on Intelligent Systems*, 403–417 (Springer).

Sundberg R, Melander E (2013) Introducing the UCDP georeferenced event dataset. *Journal of Peace Research* 50(4):523–532, https://ucdp.uu.se/downloads/index.html#ged_global.

Szymański P, Kajdanowicz T (2017) A scikit-based python environment for performing multi-label classification. arXiv abs/1702.01460, https://doi.org/10.48550/arXiv.1702.01460.

Tunstall L, Von Werra L, Wolf T (2022) *Natural Language Processing with Transformers* (O'Reilly Media, Inc., UK).

Ukjae J (2021) Smaller-LaBSE. Github. Accessed October 1, 2022, https://github.com/jeongukjae/smaller-labse.

Vargas L, Emami P, Traynor P (2020) On the detection of disinformation campaign activity with network analysis. *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, 133–146.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A, Kaiser Ł, Polosukhin I (2017) Attention is all you need. *Advances in neural information processing systems* 30, https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang Z, Mayhew S, Roth D, et al. (2019) Cross-lingual ability of multilingual BERT: An empirical study. arXiv abs/1912.07840, https://doi.org/10.48550/arXiv.1912.07840.

Warren C, Barreto A (2020) Artificial neural networks for automated detection of hostile information campaigns, unpublished report to DI2O, CODA lab, Department of Defense Analysis, Naval Postgraduate School, Monterey, CA.

Warren TC (2015) Explosive connections? Mass media, social media, and the geography of collective violence in African states. *Journal of Peace Research* 52(3):297–311.

Weber D, Neumann F (2021) Amplifying influence through coordinated behaviour in social networks. *Social Network Analysis and Mining* 11(1):1–42.

Williams EM, Novak V, Blackwell D, Platzman P, McCulloh I, Phillips NE (2020) Homophily and transitivity in bot disinformation networks. *2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 1–7 (IEEE).

Wu L, Morstatter F, Carley KM, Liu H (2019) Misinformation in social media: definition, manipulation, and detection. *ACM SIGKDD Explorations Newsletter* 21(2):80–90.

Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, et al. (2016) Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv abs/1609.08144, https://doi.org/10.48550/arXiv.1609.08144.

Xu Y, Wang C, Dan Z, Sun S, Dong F (2019) Deep recurrent neural network and data filtering for rumor detection on sina weibo. *Symmetry* 11(11):1408.

Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, Barua A, Raffel C (2020) mt5: A massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934 https://doi.org/10.48550/arXiv.2010.11934.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California