



TECHNICAL MANUAL

**EARLY WARNING
SYSTEM ON GENDER
BACKLASHES**

Contacts

UNDP Gender Team

Leona Verdadero

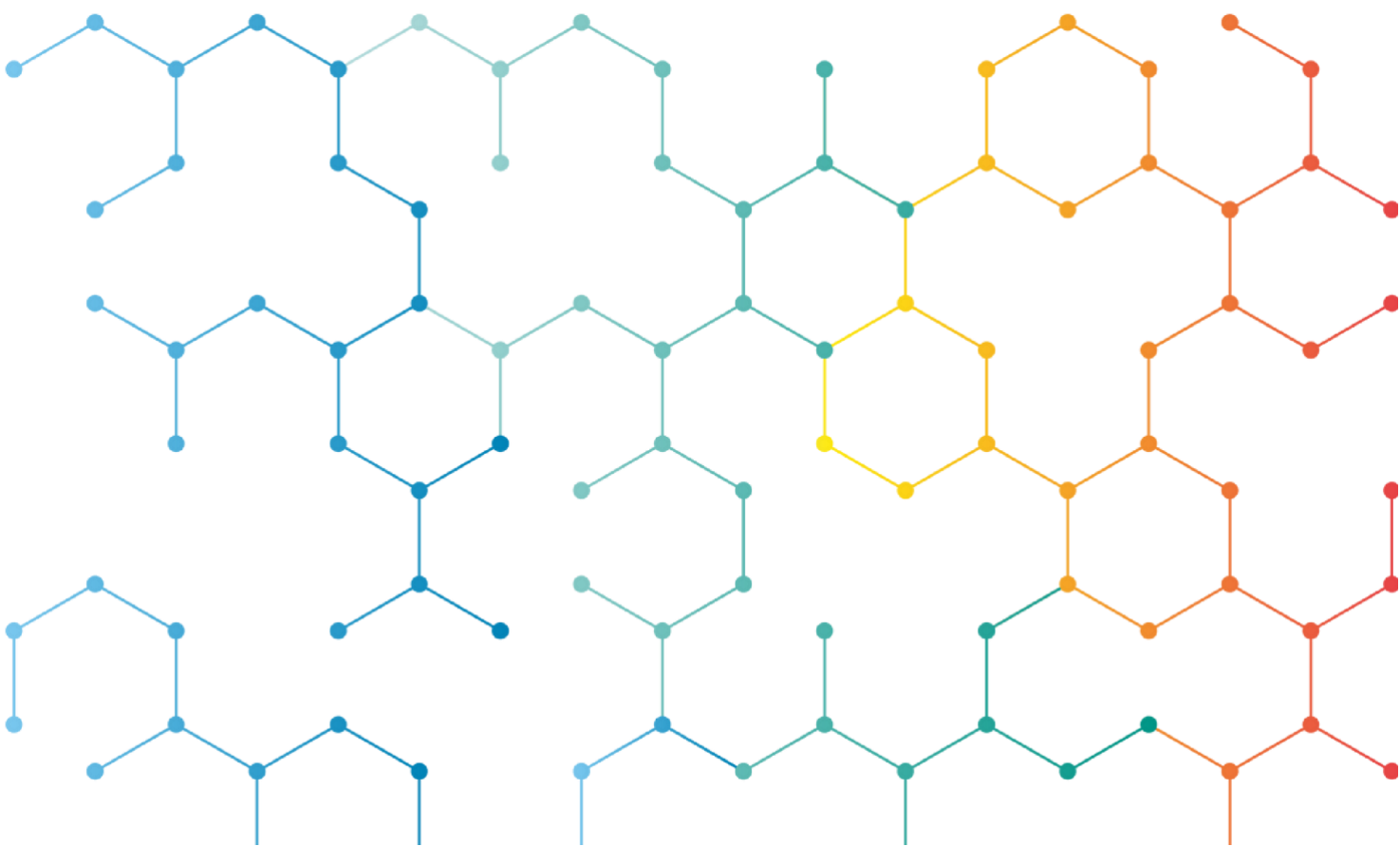
Joanna Hill

This technical manual aims to document necessary details pertaining to methodology, data architecture, models as well as possible avenues for improvements as part of the development of an Early Warning System on gender backlashes – an AI software designed for scanning social networks, social media and mass media for trends and signs of specific gender-related events, tracking public discussions and public opinions for indications of backlashes relating to gender equality and women's rights. The manual is prepared by the SDG Integration Team within UNDP Global Policy Network in collaboration with the Gender Team.



Abstract

The Early Warning System on gender backlashes is an AI software designed for scanning social networks, social media and mass media for trends and signs of specific gender related events, tracking public discussions and public opinions for indications of backlashes relating to gender equality and women's rights. The Early Warning System on gender backlashes provides an analysis of social media, and language, to understand social narratives, complaints and concerns around policies and government decisions. The software was first released in July 2022 as a starting point for discussions with the Uganda, Colombia and Philippines country offices. Second version was release in December 2022, including a fine-tuned hate speech on gender backlashes classifier.

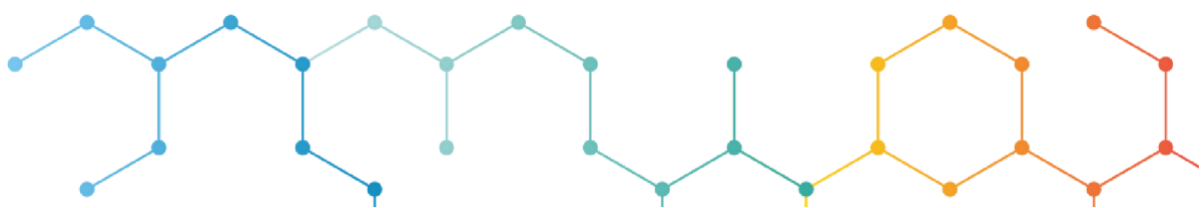


Keywords: AI, social media, hate speech on gender backlashes



Contents

1. INTRODUCTION	1
1.1 Introduction	1
1.2 Requirements	2
1.3 Development timeline phase I.....	1
1.4 Development timeline phase II	2
1.5 Software used	1
2. METHODOLOGY.....	2
1.1 Data sources.....	2
1.2 Hate speech identification.....	5
1.3 Methodology limitations.....	12
3. DATA ARCHITECTURE	14
1.1 Overview	14
1.2 Data ingestion layer	15
1.2.1 UN Global pulse.....	15
1.2.2 Twint scrapping tool.....	17
1.3 Data storage layer	19
1.3.1 Delta lake.....	19
1.4 Data analysis	23
1.4.1 Databricks.....	23
1.4.2 Hugging Face.....	28
1.4.3 Spark NLP.....	29
1.5 Data consumption layer.....	31
1.5.1 FTP SiteGround server.....	31
1.5.2 WordCloud python API.....	33
4. AI MODELS	37
1.1 Introduction.....	37
1.2 Gender classification	39
1.2.1 Dataset.....	40
1.2.2 Pretrained base model.....	42

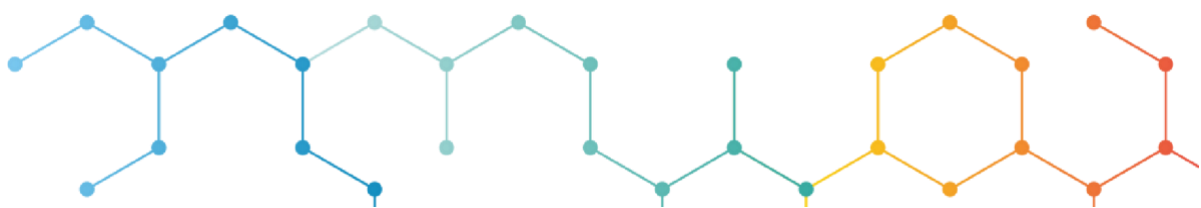




1.3 Hate speech classification	46
1.3.1 Phase I	47
1.3.2 Phase II	51
1.4 Sentiment analysis	65
1.4.1 Sentiment analysis in English	65
1.4.2 Sentiment analysis in Spanish	67
1.5 Topic modelling	68
1.5.1 Latent Dirichlet Allocation LDA	69
1.5.2 Zero-shot classification	72
5. CONCLUSIONS	75
1.1 Conclusions	75
1.2 Carbon footprint	76
6. GLOSSARY	78
7. REFERENCES	79
8. APPENDIX A: BATCH DATA CREATION CODE	81
9. APPENDIX B: CONTINUOUS DATA PIPELINE CODE	98
1.1 Phase I	98
1.2 Phase II	103
10. APPENDIX C: CONCEPT NOTE AI EWS	122



1.3.1 Translators	44
1.3.2 Hate speech	46
1.4 Sentiment analysis	48
1.4.1 Sentiment analysis in English.....	48
1.4.2 Sentiment analysis in Spanish.....	50
1.5 Topic modelling	51
1.5.1 Latent Dirichlet Allocation LDA	52
1.5.2 Zero-shot classification.....	55
5. CONCLUSIONS	57
1.1 Conclusions.....	57
1.2 Carbon footprint.....	58
1.3 Future improvements	59
6. GLOSSARY	60
7. REFERENCES	61
8. APPENDIX A: BATCH DATA CREATION CODE	63
9. APPENDIX B: CONTINUOUS DATA PIPELINE CODE	81
10. APPENDIX C: CONCEPT NOTE AI EWS	88





Introduction

1.1 Introduction

The goal of the Early Warning System on gender backlashes software is to showcase how social media can help improve the response time and contribute to effective actions in gender responsive policy measures to achieve gender equality.

Social media scanning will be used to capture public opinion trends and events regarding gender-based violence, level of aggression and evolution of the harassment, insults and hate speech. In this context, social media analysis is a powerful tool to get to know the scope, prevalence, and dynamics for gender-based violence with up-to-date and high-quality data.

The analysis will focus on presenting hate speech related discourse on the four dimensions of the World Values Survey questions used for the Global Social Norms Index (GSNI), i.e. politics, education, economic matters and physical integrity. In addition, some other issues that impact gender equality and women's rights have also been tracked, i.e. women in the workplace, women in STEM or care responsibility.

This document will present technical details regarding the implementation of the software, including:

i. Methodology

ii. Data sources (APIs vs scrapping)

iii. Ethical data management

iv. Model implementation

Implementation in a distributed environment, including multilingual gender classifiers, multilingual hate speech detection, multilingual sentiment analysis and topic modelling in several high-resource and low-resource languages

v. Monitor design

The scope of the document includes as well getting insights of the technical approach followed along the project, such as frameworks or datasets used, scalability approaches or NLP libraries and the lessons learnt.





1.2 Requirements

The scope of the first release of the Early Warning System (EWS) on gender backlashes is to provide meaningful statistics regarding hate speech on public opinion based on gender. The following requirements shall be fulfilled along the project:

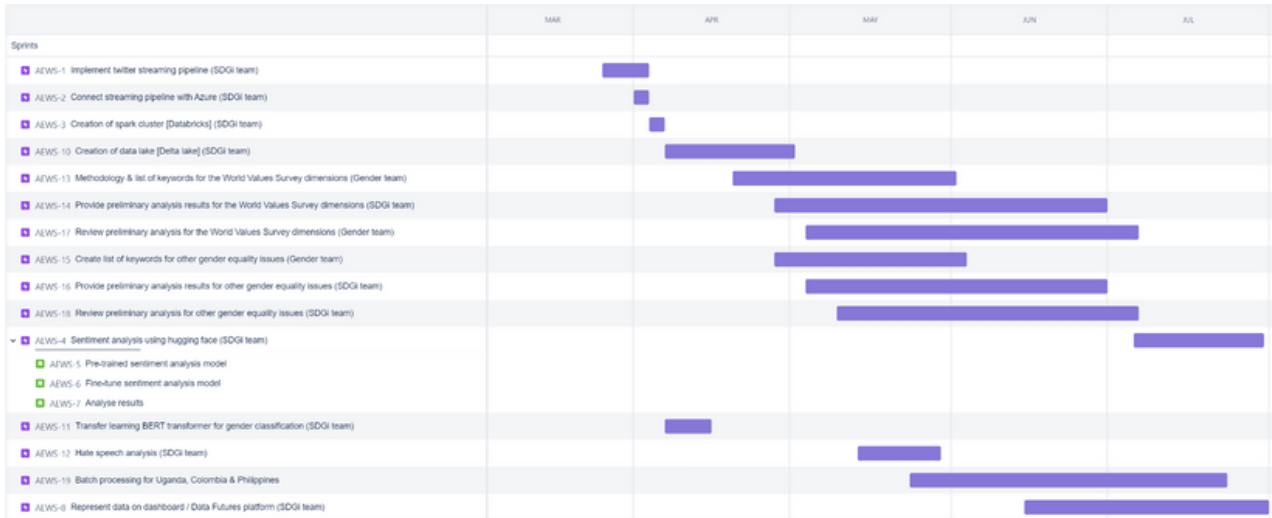
- Includes an analysis for three target countries: Colombia, the Philippines and Uganda
- Covers the following topics: education, work, politics, reproductive rights, gender-based violence
- Searches for geo-localized tweets based on keywords provided (appendix C)
- Whenever possible, use local language models to perform the analysis. If no sufficient model is found in the local language, the tweet information shall be translated into English
- Includes a gender prediction model of the Twitter user posting the information
- Includes a custom hate speech model targeted to women and girls to automatically get the intention of the user
- Includes a sentiment analysis model to automatically get the degree of positive/negative/neutral sentiment of each user comment
- Classifies each topic into subtopics using a topic modelling approach
- Analyzes Twitter historical data (batches) using the UN Global Pulse Twitter API
- Implements a continuous data pipeline, i.e. new tweets will be added to the aggregated database on a daily basis.
- Applies privacy and ethical approaches to data
- Stores all gathered data into a proprietary database





1.3 Development Timeline

1.3.1 Phase I



1.3.2 Phase II

	August 2022	September 2022	October 2022	November 2022	December 2022
Prepare dataset					
Generate & apply model to dataset					
Taxonomy created by Latin America Regional Hub					
Manual data validation (UN Volunteers)					
Dynamic search model					

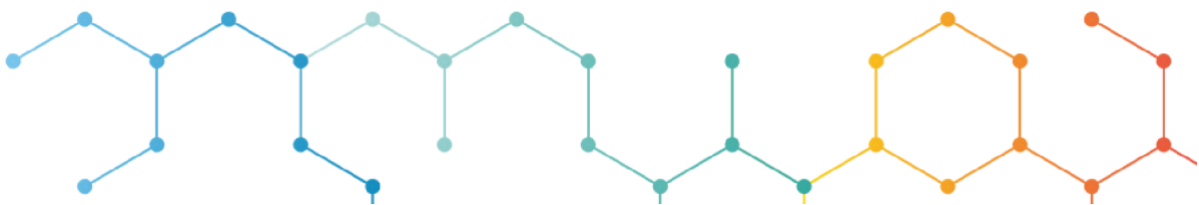




1.4 Software used

The Early Warning System on gender backlashes has been developed using the following set of tools:

1. Azure Databricks
2. Delta lake
3. Hugging-face
4. UN Global Pulse Twitter API
5. Twint
6. Tensorflow
7. Spark NLP
8. Fast API
9. Siteground server





Methodology

1.1 Data Sources

Online hate speech on social media has quadrupled in the last two years [11][12] leading to an unprecedented interest in automatic hate speech detection tools to monitor and control these kinds of attitudes against women.

The EWS Project uses Twitter data for the detection of events and trends regarding hate speech against women. Twitter has been selected as the main data source for the project because of the easiness to extract data from the social platform and the great amount of current and historical information available.

In December 2017, Twitter began enforcing new policies towards hate speech, banning multiple accounts as well as setting new guidelines for what will be allowed on their platform. Twitter has a Hateful Conduct Policy, that states "Freedom of expression means little if voices are silenced because people are afraid to speak up. We do not tolerate behavior that harasses, intimidates, or uses fear to silence another person's voice. If you see something on Twitter that violates these rules, please report it to us." Twitter's definition of hate speech ranges from "violent threats" and "wishes for the physical harm, death, or disease of individuals or groups" to "repeated and/or non-consensual slurs, epithets, racist and sexist tropes, or other content that degrades someone.". Punishments for violations range from suspending a user's ability to tweet until they take down their offensive/ hateful post to the removal of an account entirely.

Focusing this analysis to a single social media platform has certain limitations, including:

- Twitter users usually represent a few sectors of the population, making it difficult to extrapolate the results obtained by means of this analysis to the whole society;
- Online behavior tends to be more aggressive than offline;
- Twitter might not be a very representative social media platform in certain countries.

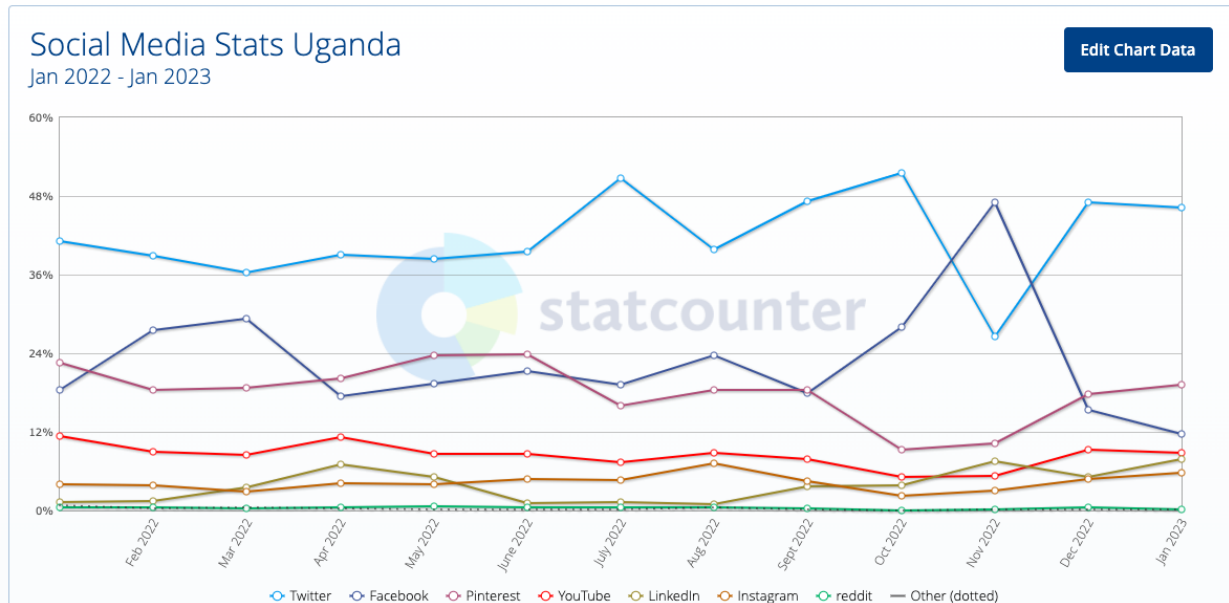
To get a sense of the relevance of this analysis for each of the countries selected by the UNDP BPPS Gender Team, statistics as of January 2023 regarding social media usage are included [1] for each country.

In Uganda, Twitter is the most popular social (January 2023):





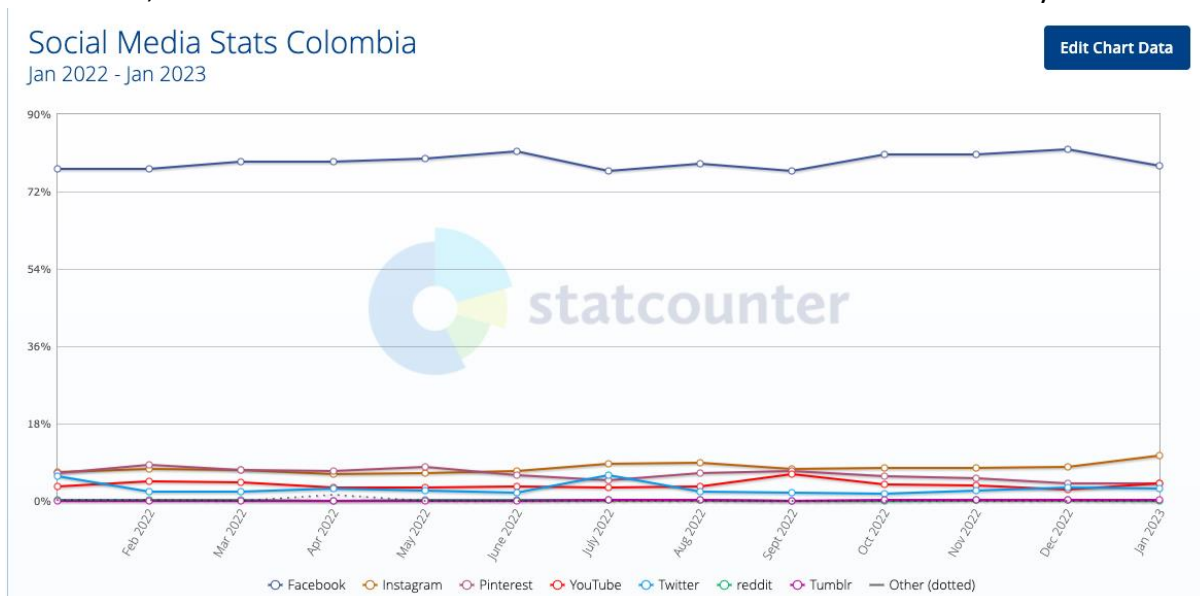
The engagement of the Ugandans in this platform has maintained steady levels versus a year ago:



In Colombia, the number of Twitter users is much lower, reaching 3% of the population. This is quite significant since Colombian users seem the most prolific ones according to our analysis.



In this case, the interest on Twitter has more or less remained stable in the last year:

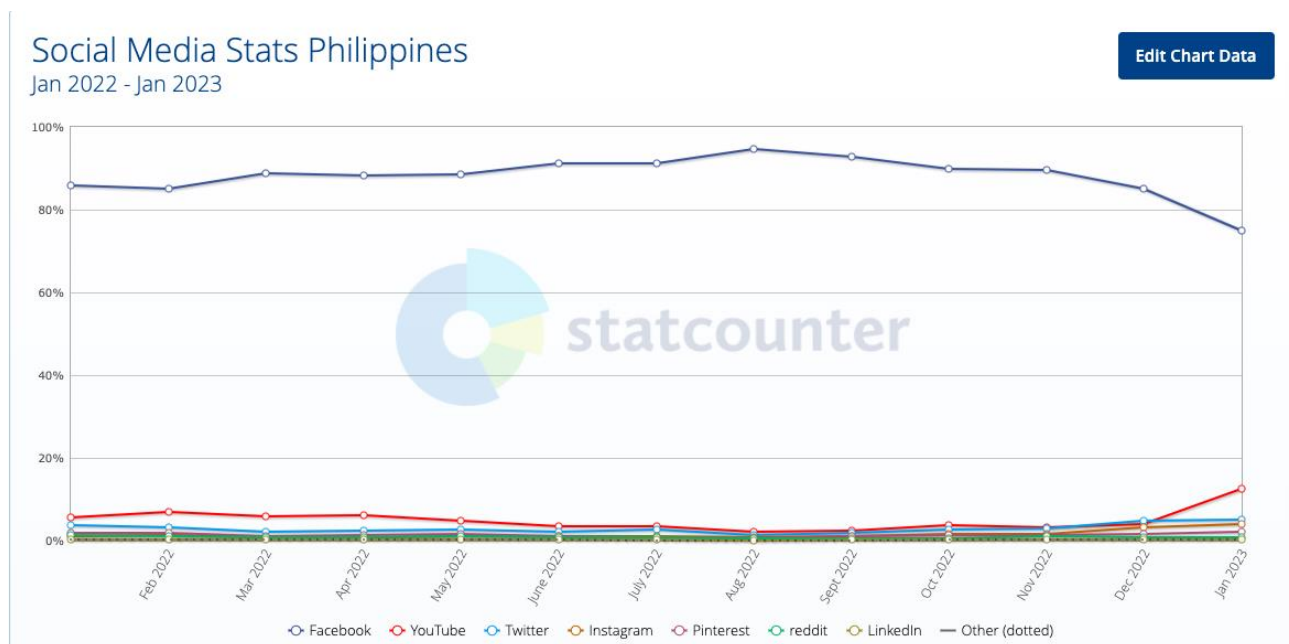




Finally, the analysis in the Philippines reveals that 5% of social media users are on Twitter, after Facebook and YouTube:



In the case of the Philippines, the engagement of the users is quite limited however it has increased in the last year.





1.2 Hate speech identification

The Twitter API provides a list of queries to implement the search of the terms/keywords provided by the UNDP BPPS Gender Team, whose concept note has been included in Appendix C.

The set of keywords gathered at the concept note has been manually extended according to the most frequent and meaningful words found in the tweets collected using the taxonomy in the concept note. This first extension has been made because of the small number of tweets that resulted from filtering tweets using just the keywords in the concept note.

A second extension of the keywords provided in the concept note was made using the grievances presented in the UN Argentinian gender tracker[1]. These terms were written in Spanish, so they have been translated into English and Tagalog to use throughout the whole analysis.

A third extension of the keywords provided in the concept note was made capturing the most relevant hate speech terms directed to women listed in the reference webpages provided by <https://hatespeechdata.com/>. This webpage includes a section specific to datasets regarding gender violence per country. The number of datasets in Spanish was very limited and no datasets were available in Tagalog.

One of the main problems of these datasets was that they did not provide single words related to hate speech. They were full tweet dialogues where the hate speech only appeared after two or three interactions. Hence, they rely on the context and this information was not useful for Twitter since it is based on just atomic word searches. The notebook containing the implementation of the keyword extraction from the webpage has been included in the zip folder provided with this memory.

The following figure shows an example of keyword extraction on the ConvAbuse dataset included in <https://hatespeechdata.com/>. All stopwords have been removed from the keyword extraction and just single words are captured.

```
keywords = kw_model.extract_keywords(text,
                                    keyphrase_ngram_range=(1, 1),
                                    stop_words='english',
                                    highlight=False,
                                    top_n=30)

keywords_list= list(dict(keywords).keys())

print(keywords_list)
```

['chatbot', 'chatbots', 'annoying', 'message', 'eliza', 'rude', 'bitch', 'dear', 'dialogue', 'messaging', 'hello', 'goodbye', 'respond', 'conversation', 'angry', 'miserable', 'hi', 'weijiehong', 'chiese', 'hanoi', 'talk', 'chan ese', 'cantmakefriendwithyou', 'talking', 'english', 'read', 'saigon', 'ignore', 'girlfriend', 'stop']

The list of the most frequent terms extracted from the ConvAbuse dataset are not very correlated to the hate speech related to gender topic address in this analysis.



A fourth extension of the keywords provided in the concept note was made using the terms suggested by UNDP volunteers labelling a custom-made hate speech dataset for the three countries.

The final search queries are 42 and correspond to the following terms:

```
query_edu_Uganda = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_stem_Uganda = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_violence_Uganda = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (rape OR sexual assault OR sexual violence OR sexual abuse OR force sex OR child marriage OR children marriage OR forced marriage OR sex trafficking OR child trafficking OR children trafficking OR female genital mutilation OR female genital cutting) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_reproductive_Uganda = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (abortion OR contraception OR birth control OR pill OR IUD OR unwanted pregnancy) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_work_Uganda = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR child OR children OR kid OR kids OR toddler OR baby OR babies OR infant OR infants OR family OR home OR domestic ) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_political_Uganda = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends OR candidate) (lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR administrations OR government OR governments OR state OR states OR political party OR political parties OR gvt OR gov OR govt OR govmnt OR govmt OR govnt OR politician OR politicians OR state funds OR party funds OR public funds OR campaign promise OR campaign promises OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresent OR misrepresented OR misrepresentation) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```





```
query_edu_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_stem_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_violence_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (rape OR sexual assault OR sexual violence OR sexual abuse OR force sex OR child marriage OR children marriage OR forced marriage OR sex trafficking OR child trafficking OR children trafficking OR female genital mutilation OR female genital cutting) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_reproductive_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (abortion OR contraception OR birth control OR pill OR IUD OR unwanted pregnancy) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_work_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR child OR children OR kid OR kids OR toddler OR baby OR babies OR infant OR infants OR family OR home OR domestic ) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_political_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR administrations OR government OR governments OR state OR states OR political party OR political parties OR gvt OR gov OR govt OR govmt OR govmnt OR govnt OR politician OR politicians OR state funds OR party funds OR public funds OR campaign promise OR campaign promises OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresent OR misrepresented OR misrepresentation) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
query_edu_Colombia = '(mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) (educación OR escuela OR escuelas OR colegio OR colegios OR educar OR educación OR universidad OR universidades OR uni OR maestro OR maestros OR profesor OR profesores OR aprendizaje OR curso OR enseñanza) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO ))'
```





query_stem_Colombia = '(mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) (STEM OR ciencia OR código OR codificar OR coding OR hacker OR coder OR tecnología OR tech OR ingeniería OR matemáticas OR tech) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'

query_violence_Colombia = '(mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) (violación OR feminicidio OR violencia vicaria OR violencia machista OR violencia de género OR asesinato OR agresión sexual OR manada OR abuso sexual OR abusos sexuales OR sexo sin consentimiento OR matrimonio infantil OR matrimonios infantiles OR matrimonio forzado OR tráfico de mujeres OR tráfico de niños OR secuestro) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'

query_reproductive_Colombia = '(mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) (aborto OR anticonceptivos OR control de natalidad OR píldora OR DIU OR embarazo no deseado) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'

query_work_Colombia = '(mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) (trabajo OR trabajando OR empleo OR carrera OR puesto OR oficina OR emplear OR empleado OR empleador OR ambición OR éxito OR fracaso OR promoción OR ascenso OR ascendido OR descenso OR relegar OR relegada OR salario OR subida salarial OR pagar OR niño OR niños OR bebé OR bebés OR menor OR menores OR familia OR casa OR hogar OR doméstico) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'

query_political_Colombia = '(mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias OR gobernadora OR candidata) (liderar OR líder OR líderes OR liderazgo OR poder OR poderoso OR política OR administración OR administraciones OR gobierno OR gobiernos OR estado OR estados OR partido político OR partidos políticos OR gobierno O gobierno O gobierno O gobierno O gobierno OR fondos estatales OR fondos del partido OR fondos públicos OR promesa electoral OR promesa de campaña OR corrupción O corrupto OR voto OR votos OR fraude OR tergiversación OR representación) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'

query_edu_hate_Colombia = '(traidora OR corrupta OR asesina OR ladrona OR cinica OR ridicula OR mentirosa OR pelotuda OR boluda OR burra OR mafiosa OR estúpida OR gorda OR bruta OR patetica OR enferma OR payasa OR inmundada OR fracasada OR siniestra OR loca OR tarada OR puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR rastrera OR estafadora OR resentida OR mugrienta OR tonta OR guarra OR zorra OR calientapollas)(educación OR escuela OR escuelas OR colegio OR colegios OR educar OR educación OR universidad OR universidades OR uni OR maestro OR maestros OR profesor OR profesores OR aprendizaje OR curso OR enseñanza) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'





```
query_stem_hate_Colombia = '(traidora OR corrupta OR asesina OR ladrona OR cinica OR  
ridicula OR mentirosa OR pelotuda OR boluda OR burra OR mafiosa OR estúpida OR gorda OR  
bruta OR patetica OR enferma OR payasa OR inmunda OR fracasada OR siniestra OR loca OR  
tarada OR puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR rastrera OR  
estafadora OR resentida OR mugrienta OR tonta OR guarra OR zorra OR calientapollas)(STEM OR  
ciencia OR código OR codificar OR coding OR hacker OR coder OR tecnología OR tech OR  
ingeniería OR matemáticas OR tech) (point_radius:[-74.063644 4.624335 25mi] OR  
place:"Bogota" OR (profile_country:CO ))'
```

```
query_violence_hate_Colombia = '(traidora OR corrupta OR asesina OR ladrona OR cinica OR  
ridicula OR mentirosa OR pelotuda OR boluda OR burra OR mafiosa OR estúpida OR gorda OR  
bruta OR patetica OR enferma OR payasa OR inmunda OR fracasada OR siniestra OR loca OR  
tarada OR puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR rastrera OR  
estafadora OR resentida OR mugrienta OR tonta OR guarra OR zorra OR calientapollas)(violación  
OR feminicidio OR violencia vicaria OR violencia machista OR violencia de género OR asesinato  
OR agresión sexual OR manada OR abuso sexual OR abusos sexuales OR sexo sin consentimiento  
OR matrimonio infantil OR matrimonios infantiles OR matrimonio forzado OR tráfico de mujeres  
OR tráfico de niños OR secuestro) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota"  
OR (profile_country:CO ))'
```

```
query_violence_hate_Colombia_ext = '(abombada OR absolutista OR agrogarca OR alcahueta  
OR altanera OR alzada OR amoral OR analfabeta OR anda a cagar OR anda a la mierda OR anda  
a terapia OR argolluda OR arpia OR arrastrada OR asesina OR asquerosa OR atorranta OR  
atrevida OR banate OR bastarda OR basura OR berreta OR bicha OR bobeta OR boluda OR  
borracha OR bosta OR braguetera OR buena para nada OR burda OR burra OR cabeza hueca OR  
cacatúa OR cachavache OR cagadora OR cagona OR calandraca OR caprichosa OR cara de OR  
casta inmunda OR catadora de waska OR cerda OR chapita OR chupasangre OR cornuda OR  
conventillera OR culo roto OR das asco OR das pena OR das repulsión OR degenerado OR gorda  
OR grosera OR gilipolla OR hacete coger OR hacete humo)(point_radius:[-74.063644 4.624335  
25mi] OR place:"Bogota" OR (profile_country:CO ))'
```

```
query_reproductive_hate_Colombia = '(traidora OR corrupta OR asesina OR ladrona OR cinica  
OR ridicula OR mentirosa OR pelotuda OR boluda OR burra OR mafiosa OR estúpida OR gorda  
OR bruta OR patetica OR enferma OR payasa OR inmunda OR fracasada OR siniestra OR loca OR  
tarada OR puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR rastrera OR  
estafadora OR resentida OR mugrienta OR tonta OR guarra OR zorra OR calientapollas)(aborto  
OR anticonceptivos OR control de natalidad OR píldora OR DIU OR embarazo no deseado)  
(point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO ))'
```

```
query_work_hate_Colombia = '(traidora OR corrupta OR asesina OR ladrona OR cinica OR  
ridicula OR mentirosa OR pelotuda OR boluda OR burra OR mafiosa OR estúpida OR gorda OR  
bruta OR patetica OR enferma OR payasa OR inmunda OR fracasada OR siniestra OR loca OR  
tarada OR puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR rastrera OR  
estafadora OR resentida OR mugrienta OR tonta OR guarra OR zorra OR calientapollas)(trabajo  
OR trabajando OR empleo OR carrera OR puesto OR oficina OR emplear OR empleado OR
```





empleador OR ambición OR éxito OR fracaso OR promoción OR ascenso OR ascendido OR descenso OR relegar OR relegada OR salario OR subida salarial OR pagar OR niño OR niños OR bebé OR bebés OR menor OR menores OR familia OR casa OR hogar OR doméstico) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'

query_work_hate_Colombia_ext = '(adoctrinada OR alienada OR anda a laburar OR autoritaria OR bajate del pony OR chamuyera OR chanta OR codiciosa OR curradora OR currera OR demagoga OR difamadora OR desubicada OR esclavista OR irrespetuosa OR ladrona OR mafiosa OR manipuladora OR mentirosa OR mercenaria OR mosquita muerta OR obtusa OR perroncha OR radicheta OR vendepatria OR vieja) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'

query_politics_hate_Colombia = '(traidora OR corrupta OR asesina OR ladrona OR cinica OR ridicula OR mentirosa OR pelotuda OR boluda OR burra OR mafiosa OR estúpida OR gorda OR bruta OR patetica OR enferma OR payasa OR inmundada OR fracasada OR siniestra OR loca OR tarada OR puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR rastrera OR estafadora OR resentida OR mugrienta OR tonta OR guarra OR zorra OR calientapollas)(liderar OR líder OR líderes OR liderazgo OR poder OR poderoso OR política OR administración OR administraciones OR gobierno OR gobiernos OR estado OR estados OR partido político OR partidos políticos OR gobierno O gobierno O gobierno O gobierno O gobierno OR fondos estatales OR fondos del partido OR fondos públicos OR promesa electoral OR promesa de campaña OR corrupción O corrupto OR voto OR votos OR fraude OR tergiversación OR representación) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'

query_politics_hate_Colombia_ext = '(antirepublicana OR maleducada OR maricona OR milf OR militanta OR mononeuronal OR ordinaria OR perversa OR psicopata OR repulsiva OR ramera OR repulsiva OR resentida OR retardada OR terrorista OR traicionera OR vibora OR ventajera OR violenta OR vomito con patas) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO))'

query_edu_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta)(edukasyon OR paaralan OR mga paaralan OR edu OR kolehiyo OR unibersidad OR guro OR pag-aaral OR kurso OR pagtuturo)'

query_stem_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta) (STEM OR hacker OR science OR code OR coding OR teknolohiya OR engineering OR matematika OR tech)'

query_violence_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR





manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta) (panggagahasa OR sekswal na pag-atake OR sekswal na karahasan OR sekswal na pang-aabuso OR puwersahin ang sex OR child marriage OR kasal ng mga bata OR forced marriage OR sex trafficking OR child trafficking OR child trafficking) '

query_reproductive_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta) (pagpapalaglag OR pagpipigil sa pagbubuntis OR birth control OR tableta OR IUD OR hindi gustong pagbubuntis) '

query_work_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta) (trabaho OR nagtatrabaho OR karera OR trabaho OR trabaho OR opisina OR trabaho OR trabaho OR trabaho OR ambisyon OR tagumpay OR kabiguan OR promosyon OR na-promote OR pagbabawas OR pagbabawas OR pagbabawas OR pagbabawas OR pagbabawas OR pagbabawas OR pagbabawas OR pagbabawas OR pagbabawas OR pagpapababa OR suweldo OR pagtaas OR pagbabayad OR anak OR bata OR bata OR bata OR sanggol OR sanggol OR sanggol OR sanggol OR sanggol OR pamilya OR tahanan OR domestic)'

query_political_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta) (pinuno OR pinuno OR pinuno OR pamumuno OR kapangyarihan OR makapangyarihan OR pulitika OR administrasyon OR mga administrasyon OR pamahalaan OR pamahalaan OR estado OR estado OR partidong pampulitika OR partidong pampulitika OR gvt OR gobyerno, OR gobyerno, OR pamahalaan, OR politiko OR mga pulitiko OR mga pondo ng estado OR mga pondo ng partido OR mga pondo ng publiko OR mga pangako sa kampanya OR mga pangako ng kampanya OR mga tiwali OR katiwalian OR bumoto OR mga boto OR pandaraya OR maling representasyon OR maling representasyon OR maling representasyon)'

query_edu_hate_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta)(batsilyer OR diploma OR dunong OR edukasyon OR estudyante OR grado OR kolehiyo OR paaralan OR pag-aaral OR pinag-aralan OR pagsusulit OR pagtuturo OR turo OR unibersidad)'

query_stem_hate_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR





manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta)(agham OR matematika OR inhinyeriya OR pag-linhinyero OR sipnayan OR teknolohiya)'

query_violence_hate_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta)(Abuso OR Pang-aabuso OR Atakeng Sekswal OR Dahas OR Karahasan OR Gahasa OR Panggagahasa OR Halay OR Panghahalay OR Insesto OR Lapastangan OR Kalapastanganan OR Lupit OR Kalupitan OR Molestiya OR Pangmomolestiya OR Pagpatay OR Puwersa OR Pamumwersa OR Sekswal na karahasan)'

query_reproductive_hate_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta)(Aborsyon OR Buntis OR Pagbubuntis OR Kontraseptibo OR Laglag OR Pagpapalaglag OR Pagdadalantao OR Pagpapaagas OR Pagpigil sa panganganak OR Pamparegla OR Panganganak)'

query_work_hate_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta)(Ambisyon OR Benepisyo OR Empleyado OR Manggagawa OR Hanapbuhay OR Karanasan OR Karera OR Kumpanya OR Opisina OR pag-angat OR pagbabawas OR Pagpapatalsik OR pagsisisante OR promosyon OR sahod OR suweldo OR tanggalan OR trabaho)'

query_political_hate_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra OR negrita OR puta OR amputa OR shuta)(Administrasyon OR balota OR boto OR eleksyon OR halalan OR kapangyarihan OR karapatan OR katungkulan OR korapsyon OR lider OR opisyal OR pagsisilbi OR pamahalaan OR gobyerno OR pamumuno OR pangangampanya OR partido OR pinuno OR pulitika OR trapo)'

query_edu_Philippines_EN = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'





```
query_stem_Philippines_EN = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'
```

```
query_violence_Philippines_EN = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (rape OR sexual assault OR sexual violence OR sexual abuse OR force sex OR child marriage OR children marriage OR forced marriage OR sex trafficking OR child trafficking OR children trafficking OR female genital mutilation OR female genital cutting) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'
```

```
query_reproductive_Philippines_EN = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (abortion OR contraception OR birth control OR pill OR IUD OR unwanted pregnancy) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'
```

```
query_work_Philippines_EN = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR child OR children OR kid OR kids OR toddler OR baby OR babies OR infant OR infants OR family OR home OR domestic ) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'
```

```
query_political_Philippines_EN = '(woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends OR candidate) (lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR administrations OR government OR governments OR state OR states OR political party OR political parties OR gvt OR gov OR govt OR govmt OR govmt OR govnt OR politician OR politicians OR state funds OR party funds OR public funds OR campaign promise OR campaign promises OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresent OR misrepresented OR misrepresentation) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'
```

```
query_edu_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'
```

```
query_stem_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'
```

```
query_violence_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore)(rape OR sexual assault
```



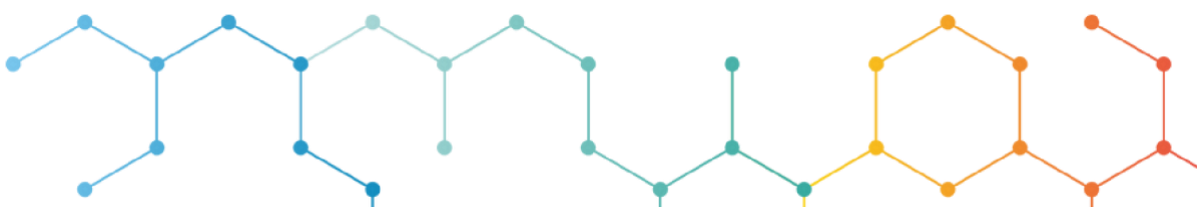


OR sexual violence OR sexual abuse OR force sex OR child marriage OR children marriage OR forced marriage OR sex trafficking OR child trafficking OR children trafficking OR female genital mutilation OR female genital cutting) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'

query_reproductive_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (abortion OR contraception OR birth control OR pill OR IUD OR unwanted pregnancy) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'

query_work_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR child OR children OR kid OR kids OR toddler OR baby OR babies OR infant OR infants OR family OR home OR domestic) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'

query_political_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore)(lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR administrations OR government OR governments OR state OR states OR political party OR political parties OR gvt OR gov OR govt OR govmnt OR govmt OR govnt OR politician OR politicians OR state funds OR party funds OR public funds OR campaign promise OR campaign promises OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresent OR misrepresented OR misrepresentation) (point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'





Data architecture

1.1 Overview

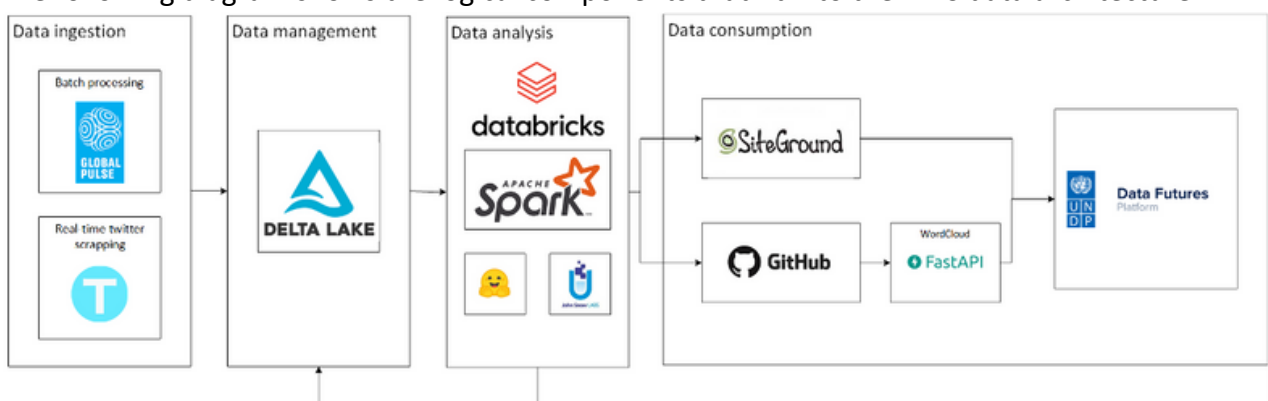
The Early Warning System on gender backlashes is based on a big data architecture to handle the ingestion, processing, and analysis of data of several gigabytes per month.

The big data architecture framework serves as a reference blueprint for big data infrastructures and solutions, logically defining how big data solutions will work, the components that will be used, how information will flow, and security details.

The EWS architecture components consists of four logical layers:

- Big Data Sources Layer: batch processing and real-time data sources.
- Management & Storage Layer: receives data from the source, converts the data into a format comprehensible for the data analytics tool, and stores the data according to its format into a delta lake.
- Analysis Layer: analytics tools extract gender prediction, hate speech classification, sentiment analysis and topic modelling from the data storage layer.
- Consumption Layer: receives results from the analysis layer and presents them to the visualization tool via FTP.

The following diagram shows the logical components that fit into the EWS data architecture:





The EWS architecture processes consist of four major processes:

- Connecting to Data Sources: connectors and adapters are capable of efficiently connecting any format of data and can connect to a variety of different storage systems, protocols, and networks.
- Data Governance: includes provisions for privacy and security, operating from the moment of ingestion through processing, analysis, storage, and deletion.
- Systems Management: highly scalable thanks to Databricks, large-scale distributed clusters are used in the EWS architecture.
- Protecting Quality of Service: Ingestion frequency and sizes.

1.2 Data ingestion layer

1.2.1 UN Global Pulse

This data source is based on a twitter API access through UN Global Pulse partnership. The API provides access to the core elements of twitter to get detailed search results. These core elements include tweets, users, direct messages, trends, media, places or spaces.

The following enterprise products have been used for the social media analysis in the Early Warning System on gender backlashes software:

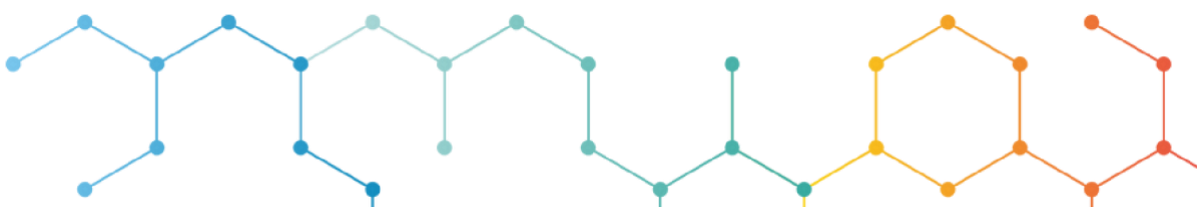
Realtime

- PowerTrack API: PowerTrack provides complete coverage of Tweets as they happen, filterable across a wide range of operators and rules

Historical data

- 30-Day Search API: The 30-Day Search API provides instant and complete access to the last 30 days of public Twitter data.
- Historical PowerTrack API: Historical PowerTrack provides access to the full archive of public Twitter data through an economical batch access process.

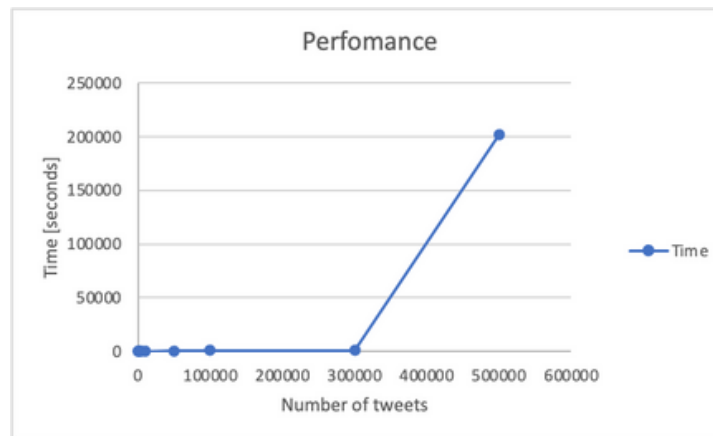
All these products must be run on the proprietary UN Global Pulse server. Any data retrieved from their platform cannot be shared in a raw format with any other platform and/or user. All the historical data for the EWS has been aggregated in coordination with the SDGi Data Visualization Analyst and pushed manually to the databricks database without any mention of userid, username or the actual tweet under analysis.





The built-in process of the batch data has consisted of applying the methodology explained in previous chapter, i.e. all the keywords detailed in section 2.1.2 and run an historical powertack search month by month. The search has been made month by month due to the following constraints:

- UN global pulse API does not currently support large amount of data retrieval (>1million tweets in a single search). This has been reported to the UN global pulse team and they are working on future improvements.
- Distributed inference process for the gender, hate speech and topic classification works better with smaller datasets for Spark. Here in a graph of the testing conducted on Spark to fine tune the number of partitions of the dataset to optimize the performance time of the inferencing process:



Typically, the number of tweets per month varies between 200.000 to 450.000 tweets. In the graph, it can be seeing that using a cluster distributed processing based on spark seems linear up to 300.000 tweets hence, it has been decided that making partitions monthly is the best approach timewise.

Once the partitions are made, a notebook hosted in the UN global pulse server retrieves the data according to the selected timeslot and the following parameters per tweet:

- Username
- Tag
- Created
- Text
- RetweetCount

The tag parameter has been used to reduce the amount of searches on historical data. This workaround has been taken because there is a maximum quota on UN global pulse twitter API of 100 days/month due to the platform pricing. As there was a requirement for the EWS to get data for a whole year, and each query sums N days even if the same dates are queried, the tag label





permits to consume just a single day containing each country and topic so the information can be properly grouped afterwards. The following tags have been defined:

- query_edu_ "Country"
- query_stem_ "Country"
- query_violence_ "Country"
- query_reproductive_ "Country"
- query_work_ "Country"
- query_political_ "Country"

The batch data creation code has been included at appendix A and in the source code zipped file provided with this memory.

1.2.2 Twint scrapping tool

Twint is an advanced Twitter scraping tool written in Python that allows for scraping Tweets from Twitter profiles without using Twitter's API. This tool has been used for several research studies in the last years, including the work of Riofrio, C. E et al[1], Fantinuoli, C[2] or Han, B., Cook, P., & Baldwin, T [3].

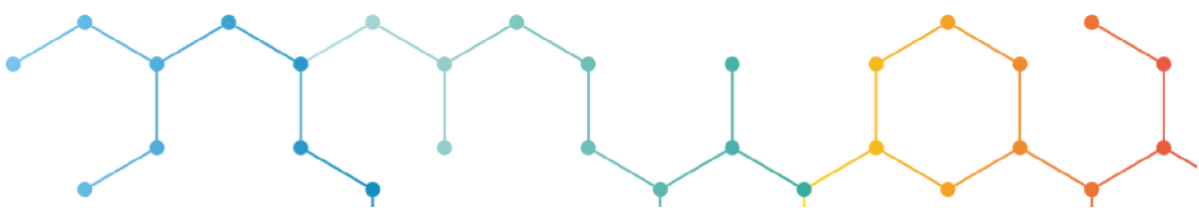
Twint utilizes Twitter's search operators to let you scrape Tweets from specific users, scrape Tweets related to certain topics, hashtags & trends, or sort out sensitive information from Tweets like e-mail and phone numbers.

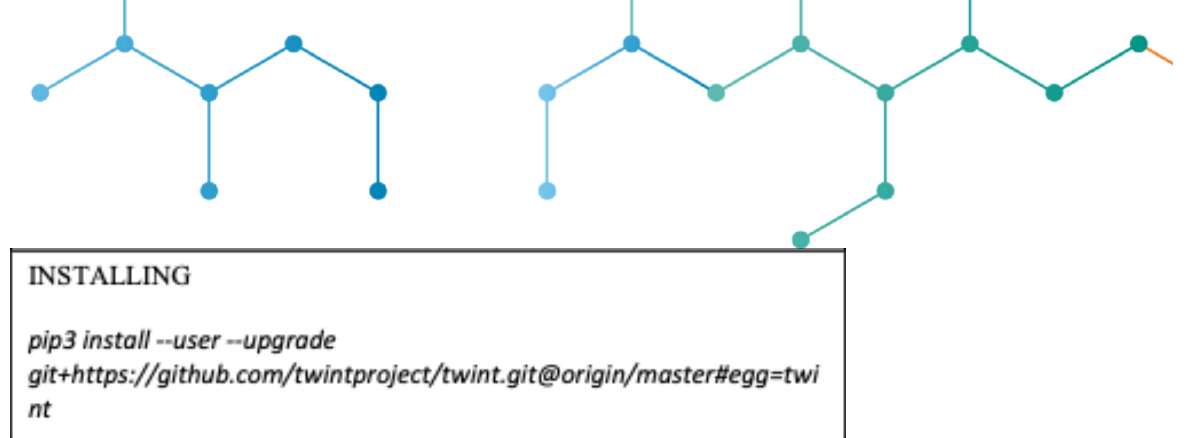
Twint also makes special queries to Twitter allowing you to also scrape a Twitter user's followers, Tweets a user has liked, and who they follow without any authentication, API, Selenium, or browser emulation.

Some of the benefits of using Twint vs Twitter API:

- Can fetch almost all Tweets (Twitter API limits to last 3200 Tweets only)
- Fast initial setup
- No rate limitations

Twitter limits scrolls while browsing the user timeline. This means that with Profile or with Favorites you will be able to get ~3200 tweets.





The same search keywords used in UN global pulse API have been used to scrap data via Twint. Just as UN global pulse historical Powertrack API was our preferred solution to create the batch dataset, Twint is the preferred tool to get continuous data daily. This decision was taken because there are no restrictions regarding the integration with the Databricks platform, we have a scheduled job on databricks that gets executed daily and get all the tweets for the day using Twint with the following configuration:

Parameter	Usage
Search	Query to search
Since	Filter Tweets sent since date
Pandas = true	Save Tweets in a DataFrame (Pandas) file. Use twint.storage.panda.Tweets_df to get dataframe
Geocode ³	Search for geocoded Tweets.

Here in a code example where these parameters are used to retrieve data for each location and topic. The whole Notebook can be found at appendix B and in the source code zipped file provided with this memory.

[1] Coordinates have been used instead of 'near' or 'location' parameters since these seem to fail to localize the tweets

```
config = twint.Config()  
config.Search = scrapList[topic]['query']  
config.Since = today  
config.Pandas = True  
twint.run.Search(config)  
df_twint = twint.storage.panda.Tweets_df
```





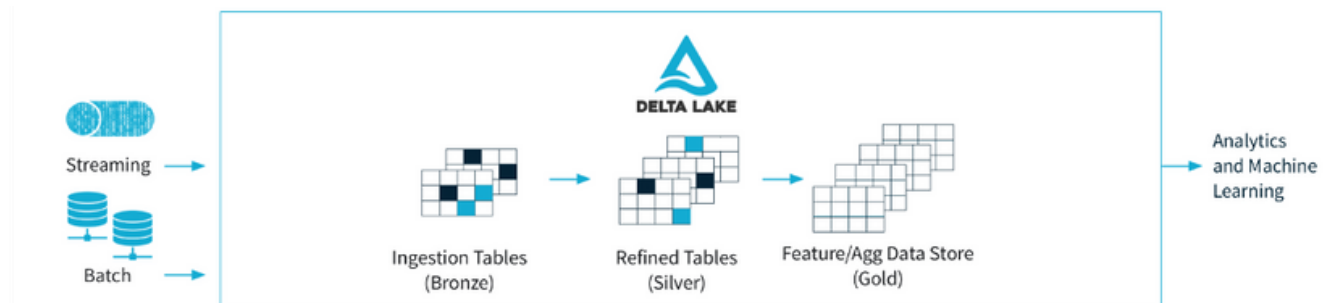
1.3 Data storage layer

1.3.1 Delta lake

All the data gathered along the project is being stored on a data lake. A data lake is a storage repository that holds a vast amount of raw data in its native format until it is needed for analytics applications. While a traditional data warehouse stores data in hierarchical dimensions and tables, a data lake uses a flat architecture to store data, primarily in files or object storage. That gives users more flexibility in data management, storage and usage.

Databricks is natively integrated with the Delta Lake solution, which has been our preferred storage framework since it has been already used and deployed in different SDGi projects using Databricks.

Delta Lake is an open-source storage framework that enables building a Lakehouse architecture with compute engines including Spark, PrestoDB, Flink, Trino, and Hive and APIs for Scala, Java, Rust, Ruby, and Python.



Specifically, Delta Lake offers:

- ACID transactions on Spark: Serializable isolation levels ensure that readers never see inconsistent data.
- Scalable metadata handling: Leverages Spark distributed processing power to handle all the metadata for petabyte-scale tables with billions of files at ease.
- Streaming and batch unification: A table in Delta Lake is a batch table as well as a streaming source and sink. Streaming data ingest, batch historic backfill, interactive queries all just work out of the box.





- Schema enforcement: Automatically handles schema variations to prevent insertion of bad records during ingestion.
- Time travel: Data versioning enables rollbacks, full historical audit trails, and reproducible machine learning experiments.
- Upserts and deletes: Supports merge, update and delete operations to enable complex use cases like change-data-capture, slowly-changing-dimension (SCD) operations, streaming upserts, and so on.

Delta Lake is a widely spread solution used in many companies such as Tencent, COMCAST, Alibaba, viacom or ciena[1].

For more information on the benefits of using the Delta Lake solution, refer to the Lakehouse whitepaper[4].

In the EWS project, three different delta lake tables have been defined for each country in the Databricks platform:

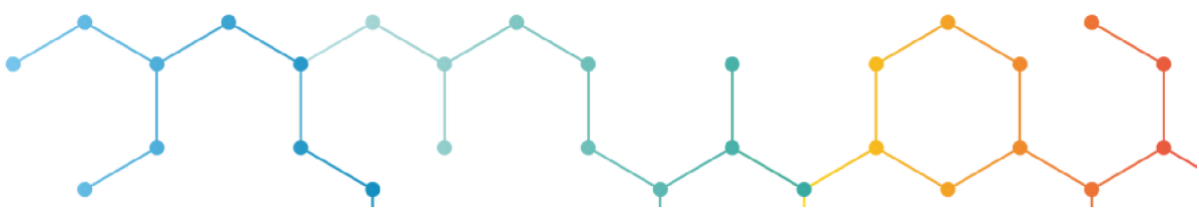
1. Bronze_”Country” table: Includes all the raw data retrieved through scrapping. It shall be noted that since this information cannot be legally provided for the tweets gathered via UN global pulse, thus this table is not available for historical data.
2. Silver_”Country” table: Includes all the predictions made based on each tweet regarding gender, subtopic , hate speech and sentiment analysis along with the following columns enherited from the corresponding bronze table:

- Username
- Date & time
- Tweet text
- Number of retweets
- Location

[1] <https://delta.io/>

Again, it shall be noted that since this information cannot be legally provided for the tweets gathered via UN global pulse, thus this table is not available for historical data.

3. Golden_”Country” table: Aggregated table based on the Silver_”Country” table containing the following information relevant to the visualizations in the data futures platform:





- Hour
- Date
- Topic
- Number of tweets
- Male
- MaleHate
- FemaleHate

Since the aggregated anonymized form of the data retrieved from the UN global Pulse can be exported to other platforms, this table is available for both historical and streaming data.

The creation of a Delta Lake table in Databricks is very straightforward, with a simple source code implementation:

```
table_name = "golden_[Country]"
```

```
write_format = 'delta' save_path =
```

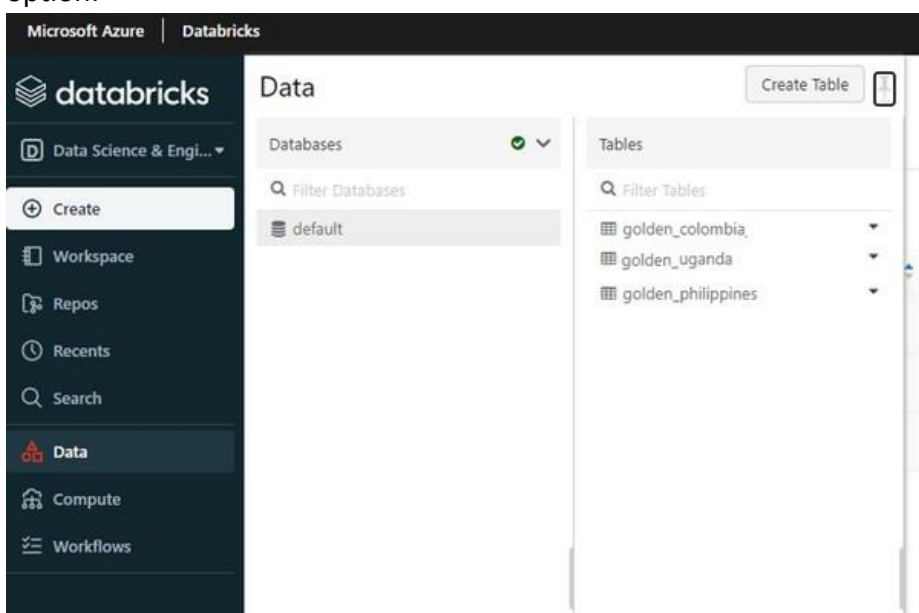
```
""path"/EarlyWarningSystem/{}".format(table_name)
```

```
spark.sql("DROP TABLE IF EXISTS default.{}".format(table_name))
```

```
result_csv.write.format(write_format) .save(save_path)
```

```
# Create the table. spark.sql("CREATE TABLE " + table_name + " USING DELTA LOCATION '" +  
save_path + "'")
```

All the delta lake tables can be consulted at the Databricks UI inside the “Data” option:





The Delta Lake tables are updated on a daily basis fed by the tweets retrieved via Twint. When writing new entries to the tables, the ACID transactions prevent data duplication in case the same input was already stored on a previous event. This guarantees data consistency on the long term and prevents memory misuse.

The following code shows a simple method from updating the Delta Lake table:

```
table_name = "golden_[Country]" save_path =  
"/path"/delta/{}'.format(table_name)  
  
sparkDF.write.format("delta").mode("append").save(save_path)  
sparkDF.write.format("delta").mode("append").saveAsTable(table_name)
```

Reading transactions on the EWS Delta Lake are also implemented since the information has to be transmitted completely (starting July 2021) to the SiteGround server via FTP. Alongside the EWS streaming pipeline, all information is processed as a spark dataframe since databricks ecosystem is built by the same team that created the spark framework and combining the two technologies give the best optimized results in terms of performance. Delta Lake reading process supports data conversions to the most popular formats, i.e., pandas dataframes, spark dataframes, parquet, etc.

```
spark.table("default.golden_[Country]") spark.read.format("delta").load("/path"/delta/...)
```

It shall be noted that all the format translations are very slow since they are operations that cause a shuffle. This means that the operations are performed atomically on each row on the table and the time consumption is very high. To prevent a failure in these operations or a failure in the Databricks configured clusters, each delta lake has to properly configure a rule for partitions, so the table is processed by chunks and the processing time significantly decreases.

The following code implements rules to make partitions based on the month variable, as already analysed in section 5.2.1.

```
df.write.option("header",True).partitionBy("Month").mode("overwrite")\  
.csv("/path"/delta)
```



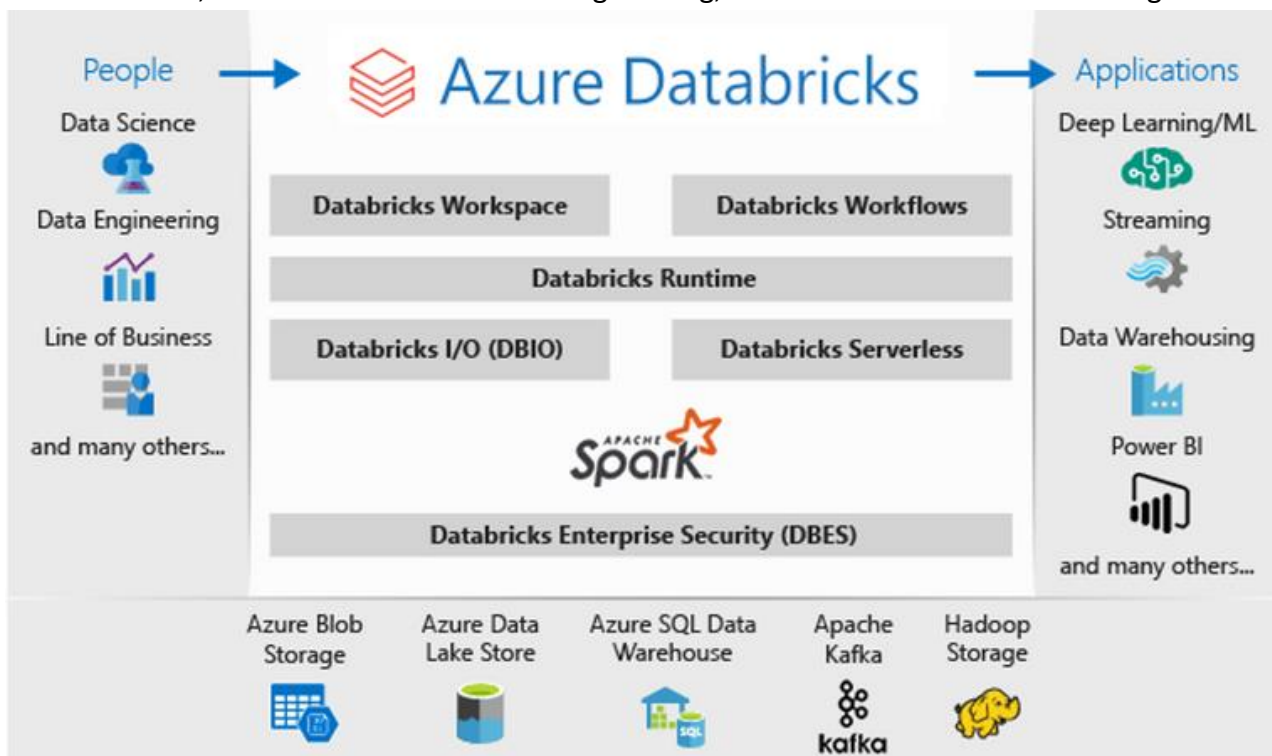


1.4 Data analysis

1.4.1 Databricks

The modelling and processing for the Early Warning System on gender backlashes have been carried out on Databricks Data Science & Engineering (sometimes called simply "Workspace") application, which is an analytics platform based on Apache Spark. It is integrated with Azure to provide a one-click setup, streamlined workflows, and an interactive workspace.

Azure Databricks is a data analytics platform optimized for the Microsoft Azure cloud services platform. Azure Databricks offers three environments for developing dataintensive applications: Databricks SQL, Databricks Data Science & Engineering, and Databricks Machine Learning.



Azure Databricks provides several options when you create and configure clusters to help you get the best performance at the lowest cost.

Following the guidelines at Azure Best practices for cluster configuration documentation[1], three different clusters have been configured inside the Databricks environment to create and run all the models used in the application.

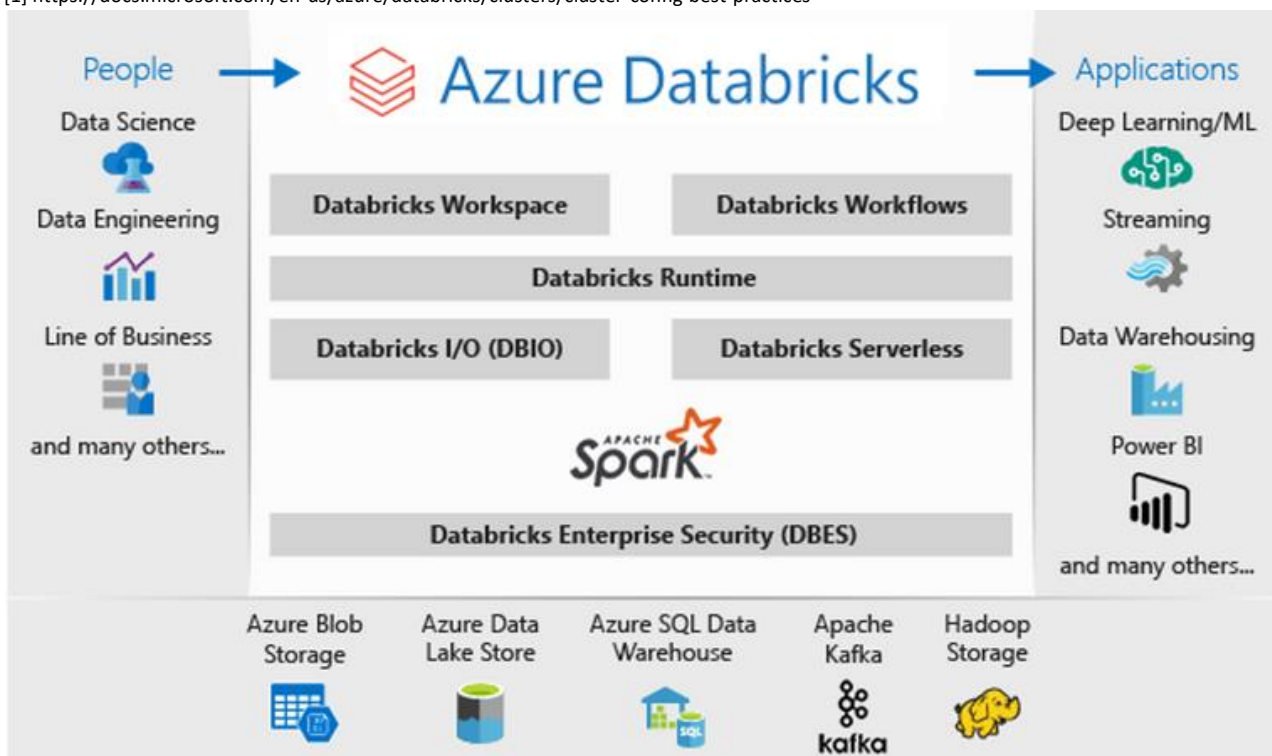




The first cluster “Analysis_GPU” has been used whenever a high-demanding resource task has been required, i.e. intensive data processing with a large amount of tweets, BERT model fine-tuned for gender prediction based on tweets or data lake management.

Next figure shows the configuration used for this cluster:

[1] <https://docs.microsoft.com/en-us/azure/databricks/clusters/cluster-config-best-practices>



Azure Databricks provides several options when you create and configure clusters to help you get the best performance at the lowest cost.

Following the guidelines at Azure Best practices for cluster configuration documentation[1], three different clusters have been configured inside the Databricks environment to create and run all the models used in the application.

The first cluster “Analysis_GPU” has been used whenever a high-demanding resource task has been required, i.e. intensive data processing with a large amount of tweets, BERT model fine-tuned for gender prediction based on tweets or data lake management.

Next figure shows the configuration used for this cluster:





[1]
<https://docs.microsoft.com/en-us/azure/databricks/clusters/cluster-config-best-practices>

Analysis_GPU

[Configuration](#) [Notebooks \(0\)](#) [Libraries](#) [Event log](#) [Spark UI](#) [Driver logs](#) [Metrics](#) [Apps](#) [Spark cluster](#)

Cluster mode ⓘ

Standard ▼

Databricks Runtime Version

10.4 LTS ML (includes Apache Spark 3.2.1, GPU, Scala 2.12)

☐ Use Photon Acceleration ⓘ

Preview

Autopilot options

☒ Enable autoscaling ⓘ

☒ Terminate after 120 minutes of inactivity ⓘ

Worker type ⓘ

Standard_NC12 112 GB Memory, 2 GPUs

Min workers 2

Max workers 8

☐ Spot instances ⓘ

Driver type

Standard_NC12 112 GB Memory, 2 GPUs

DBU / hour: 9 - 27 ⓘ

Standard_NC12

It shall be noted that autoscaling has been enabled on the cluster configuration. Autoscaling allows clusters to resize automatically based on workloads. Autoscaling can benefit many use cases and scenarios from both a cost and performance perspective.

The second cluster “Analysis_GPU” is a slightly smaller cluster design to manage and store high-volume datasets that have been processed on the UN Global Pulse server. These clusters have also been useful to validate models on different datasets and create visualizations.

Next figure shows the configuration used for this cluster:



Analysis_GPU

Configuration Notebooks (0) Libraries Event log Spark UI Driver logs Metrics Apps Spark cluster

Cluster mode ?

Standard

Databricks Runtime Version

10.4 LTS ML (includes Apache Spark 3.2.1, GPU, Scala 2.12)

☐ Use Photon Acceleration ? Preview

Autopilot options

☒ Enable autoscaling ?

☒ Terminate after 120 minutes of inactivity ?

Worker type ?

Standard_NC12

112 GB Memory, 2 GPUs

Min workers Max workers

2

8

☐ Spot instances ?

Driver type

Standard_NC12


112 GB Memory, 2 GPUs

DBU / hour: 9 - 27 ?

Standard_NC12

The third cluster "Analysis_CPU" is a smaller cluster design to run the POC of the Early Warning System on gender backlashes and process datasets with a number of tweets less or equal to 50000.

Next figure shows the configuration used for this cluster:



Analysis_CPU

[Configuration](#) [Notebooks \(0\)](#) [Libraries](#) [Event log](#) [Spark UI](#) [Driver logs](#) [Metrics](#) [Apps](#) [Spark cluster U](#)

Cluster mode

Standard

Databricks Runtime Version

9.1 LTS (includes Apache Spark 3.1.2, Scala 2.12)

☐ Use Photon Acceleration [Preview](#)

Autopilot options

☒ Enable autoscaling

☒ Terminate after minutes of inactivity

Worker type

Standard_DS3_v2

14 GB Memory, 4 Cores

Min workers Max workers

2

8

☐ Spot instances

Driver type

Standard_DS3_v2

14 GB Memory, 4 Cores

DBU / hour: 2.25 - 6.75

Standard_DS3_v2

Every analysis and visualization implemented on Azure Databricks has been included in a Notebook, hosted at the UNDP workspace inside maria.saiz.munoz@undp.org user. All these notebooks have been included in the zip folder delivered with this memory.

A notebook is a web-based interface to a document that contains runnable code, visualizations, and narrative text. Each notebook shall be attached to one of the previously defined clusters to execute its code.

Every notebook runs on top of the Databricks runtime, that features the following components used inside our project:

- Apache Spark
- Delta lake
- Preinstalled python and scala libraries
- Ubuntu and its accompanying system libraries
- Jobs to schedule periodic tasks





The following custom libraries have been additionally installed on the clusters to perform the analysis:

- Spark NLP
- Twint
- Nest_asyncio
- NLTK
- Transformers

To implement the continuous pipeline, a job has been created on Azure databricks. This job starts a cluster everyday at 6.00AM and scraps Twitter, classifies each tweet according to the different models trained, sends via FTP all the aggregated data as a .csv file and uploads all the tweets to github so information is available for the WordCloud API.

The job workflow can be checked daily at the following Databricks path:

Start time	Run ID	Launched	Duration	Spark	Status	Actions
Jul 26 2022, 6:00 AM CEST	10742	By scheduler	29m 20s	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 25 2022, 6:00 AM CEST	17172	By scheduler	24m 11s	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 24 2022, 6:00 AM CEST	16780	By scheduler	12m 24s	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 23 2022, 6:00 AM CEST	16266	By scheduler	29m 38s	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 22 2022, 6:00 AM CEST	15204	By scheduler	28m 56s	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 21 2022, 6:00 AM CEST	14114	By scheduler	34m 5s	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 20 2022, 6:00 AM CEST	12834	By scheduler	32m 1s	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 19 2022, 6:00 AM CEST	11649	By scheduler	29m 37s	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 18 2022, 6:00 AM CEST	11105	By scheduler	35 m	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 17 2022, 6:00 AM CEST	9607	By scheduler	41m 27s	Spark UI / Logs / Metrics	✓ Succeeded	
Jul 16 2022, 6:00 AM CEST	8428	By scheduler	2h 22m 14s	Spark UI / Logs / Metrics	✓ Succeeded	

1.4.2 Hugging Face

Hugging Face [1] is a community and data science platform that provides:

- Tools that enable users to build, train and deploy ML models based on open source (OS) code and technologies.

[1] <https://huggingface.co/>





- A place where a broad community of data scientists, researchers, and ML engineers can come together and share ideas, get support and contribute to open-source projects. Most Hugging Face's community contributions fall under the category of NLP (natural language processing) models.

One of the major advantages of using Hugging Face's tools is that you can reduce training time, resources, and environmental impact of creating and training a model from scratch. By fine tuning an existing pre-trained model rather than training everything from scratch you can get from data to predictions in a much shorter space of time.

INSTALLING

```
pip install git+https://github.com/huggingface/transformers
```

1.4.3 Spark NLP

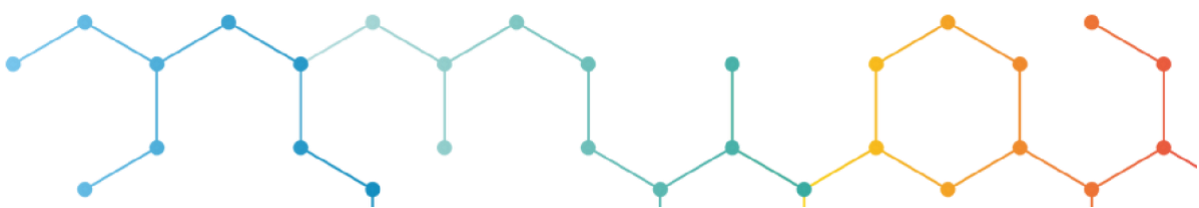
Spark NLP[1] is an open-source text processing library for advanced natural language processing for the Python, Java and Scala programming languages. The library is built on top of Apache Spark and its Spark ML library.

Its purpose is to provide an API for natural language processing pipelines that implements recent academic research results (transformers such as BERT, CamemBERT, ALBERT, ELECTRA, XLNet, DistilBERT, RoBERTa, DeBERTa, XLM-RoBERTa, Longformer, ELMO, Universal Sentence Encoder, Google T5, MarianMT, and OpenAI GPT2) as productiongrade, scalable, and trainable software. The library offers pre-trained neural network models, pipelines, and embeddings, as well as support for training custom models.

The design of the library makes use of the concept of a pipeline which is an ordered set of text annotators. Out-of-the-box annotators include, tokenizer, normalizer, stemming, lemmatizer, regular expression, TextMatcher, chunker, DateMatcher, SentenceDetector, DeepSentenceDetector, POS tagger, ViveknSentimentDetector, sentiment analysis, named entity recognition, conditional random field annotator, deep learning annotator,

[1] <https://nlp.johnsnowlabs.com/>

spell checking and correction, dependency parser, typed dependency parser, document classification, and language detection.





Here is an example of an annotator used in the EWS system to speed up the inference of hate speech in English: `document_assembler = DocumentAssembler() \`

```
.setInputCol('text') \
.setOutputCol('document')
```

```
tokenizer = Tokenizer() \ .setInputCols(['document'])
```

```
\
.setOutputCol('token')
```

```
sequenceClassifier = BertForSequenceClassification \
.pretrained('bert_sequence_classifier_dehatebert_mono', 'en') \
.setInputCols(['token', 'document']) \
.setOutputCol('class') \
```

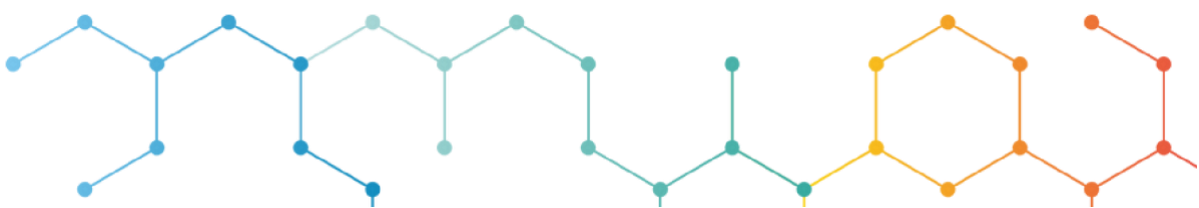
```
.setCaseSensitive(True) pipeline =
```

```
Pipeline(stages=[
    document_assembler,
    tokenizer,  sequenceClassifier
])
```

```
result = pipeline.fit(res).transform(res)
```

The full code is available at [TweetScrapping.ipynb](#) notebook in the zip folder included in this memory.

The Models Hub is a platform for sharing open source as well as licensed pre-trained models and pipelines, such as the 'bert_sequence_classifier_dehatebert_mono' included in the code example above. Models Hub includes pre-trained pipelines with tokenization, lemmatization, part-of-speech tagging, and named entity recognition exist for more than thirteen languages; word embeddings including GloVe, ELMo, BERT, ALBERT, XLNet, Small BERT, and ELECTRA; sentence embeddings including Universal Sentence Embeddings (USE) and Language Agnostic BERT Sentence Embeddings (LaBSE). It also includes resources and pre-trained models for more than two hundred languages. Spark NLP base code includes support for East Asian languages such as tokenizers for Chinese, Japanese, Korean; for right-to-left languages such as Urdu, Farsi, Arabic, Hebrew and pretrained multilingual word and sentence embeddings such as LaUSE and a translation annotator.





Spark NLP is licensed under the Apache 2.0 license. The source code is publicly available on GitHub. Prebuilt versions of Spark NLP are available in PyPi and Anaconda Repository for Python development, in Maven Central for Java & Scala development, and in Spark Packages for Spark development. These are the Spark NLP library versions (PyPi and Maven) installed in the Databricks clusters used in the EWS project:

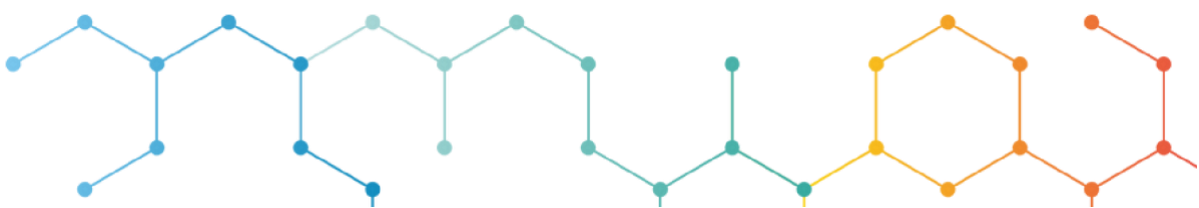
<input type="checkbox"/>	Name	Type	Status
<input type="checkbox"/>	com.johnsnowlabs.nlp:spark-nlp-gpu_2.12:4.0.0	Maven	🟢 Installed
<input type="checkbox"/>	spark-nlp==4.0.0	PyPI	🟢 Installed

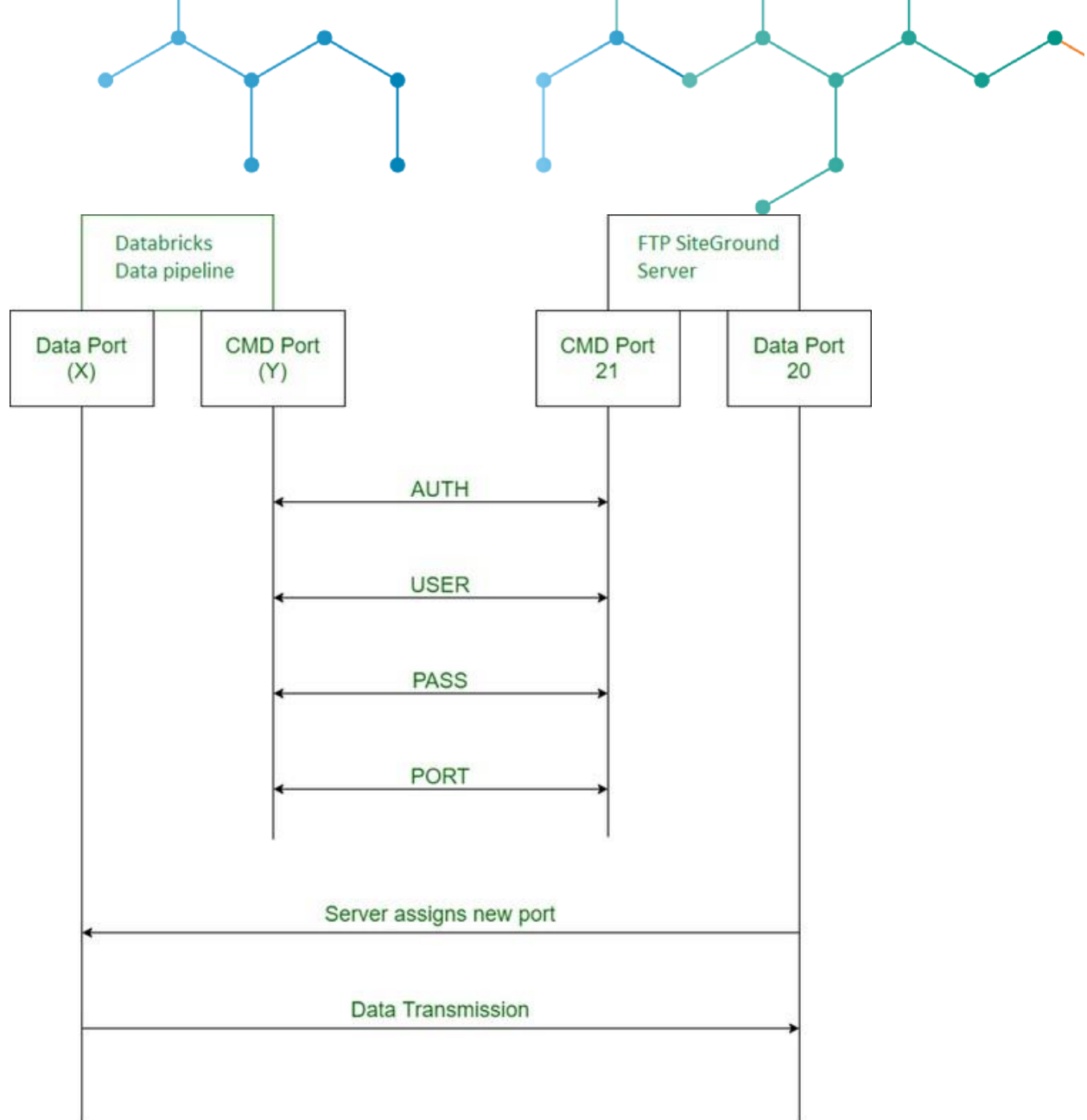
1.5 Data consumption layer

1.5.1 FTP SiteGround server

Once all the data has been cleaned-up, classified and stored, it has to be delivered to the data futures platform, where all the visualizations will be managed.

For that purpose, the aggregated datasets are sent via FTP to a UN SiteGround FTP server, where all the data is stored for the different countries and read by the data futures platform.





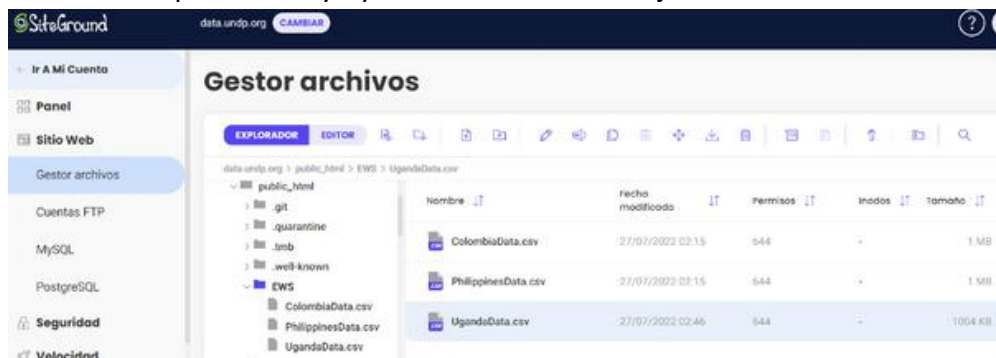
An FTP client python library called `ftplib` has been installed on the Databricks TweetScraping.ipynb notebook. The aggregated data has been pushed to the server as a .csv file but a function to deliver the same data in a .json format has also been coded:

```
def sendjson(df,fileName):
    ftp = FTP('hostname')  ftp.login("username",
    "password")
    ftp.cwd('data.undp.org/public_html/EWS')
    #ftp.cwd('staging2.data.undp.org/EWS')
    df.to_json(fileName,orient='records', indent=4)
    with open(fileName, "rb") as f:
        ftp.storbinary('STOR ' + os.path.basename(fileName), f)  ftp.quit()
```




```
def
sendcsv(df,fileName): ftp = FTP('hostname')
ftp.login("username", "password")
ftp.cwd('data.undp.org/public_html/EWS')
df.to_csv(fileName,index=False) with
open(fileName, "rb") as f:
    ftp.storbinary('STOR ' + os.path.basename(fileName), f)
    ftp.quit()
```

The FTP server stores the information in the folder pointed by the `cwd` command, where the Data Futures Platform gets the data from. This folder contains three .csv file, one for each country. The content of each file is updated daily by the Azure Databricks job described in section 5.4.1.



1.5.2 WordCloud python API

Amongst the visualizations of the EWS, one of the most relevant tools regarding the NLP analysis made over the social media data analysis is the most frequent words presented on tweets for a specific period of time, i.e. the so-called word cloud.

A word cloud, or tag cloud, is a textual data visualization which allows anyone to see in a single glance the words which have the highest frequency within a given body of text. Word clouds are typically used as a tool for processing, analyzing, and disseminating qualitative sentiment data.

Deploying a word cloud is a high resource-demanding task that should not be hosted on the front-end of a web application to prevent high latencies in the visualizations.

Due to this constraint, the word cloud has been deployed as a python micro-service hosted on the Heroku platform. Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. Then, the data futures platform makes a request to the python API hosted in Heroku and the API response is delivered via HTTP protocol to the data







The key features are:

- Fast: Very high performance, on par with NodeJS and Go (thanks to Starlette and Pydantic). One of the fastest Python frameworks available.
- Fast to code: Increase the speed to develop features by about 200% to 300%. *
- Fewer bugs: Reduce about 40% of human (developer) induced errors. *
- Intuitive: Great editor support. Completion everywhere. Less time debugging.
- Easy: Designed to be easy to use and learn. Less time reading docs.
- Short: Minimize code duplication. Multiple features from each parameter declaration.
- Fewer bugs.
- Robust: Get production-ready code. With automatic interactive documentation.
- Standards-based: Based on (and fully compatible with) the open standards for APIs: OpenAPI (previously known as Swagger) and JSON Schema.

This fastapi API based on python has been used to perform all the calculations on the selected tweets and provide a meaningful summarization of the most frequent words, that will be pulled to the DFP using gunicorn tool.

1.5.2.1.2 Installation

To support fast API capabilities, the following python libraries shall be installed:

`pip install fastapi pip install`

“uvicorn[standard]” **1.5.2.1.3**

Implementation

Fast API framework requires a predefined set of files located in the main directory. The file structure shall always contain the following files:

- 1.main.py - Main application progeam
- 2.Procfile – Command to run uvicorn on a server
- 3.Requirements – List of python packages dependencies
- 4.Runtime – Fast API runtime

1.5.2.2.1 Deployment

The fast API can be deployed in almost any web hosting service available nowadays. For the sake of simplicity, Heroku platform has been chosen due to its prebuilt integration with fast API.

To start the micro service on the host, the following command shall be used:

`uvicorn main:app --reload`





Then, the API can be accessed remotely from the data futures platform to query different frequency lists depending on the query parameters.

1.5.2.2 Functional implementation

The word cloud API is based on the WordCloud python library[1], that simplifies the process of removing stopwords, create word clouds using expressions of N words and optimize the word frequency calculation.

The main application source code can be found at appendix C. It shall be noted that to customize the set of stopwords per country, spaCy library has been used to get a set of English, Spanish and Tagalog stop words.

The API returns a list of the 50 most frequent words in the input text with its frequency values.

[1] <https://pypi.org/project/wordcloud/>

AI models

1.1 Introduction

The EWS shall classify each tweet according to different categories. These categories have been selected during the methodology refinement and are updated for every new tweet on a daily basis using a streaming pipeline. For EWS v1.0, for the three countries, the inference phase does not take longer than 40 minutes per day. For EWS v2.0, the inference phase does not take longer than 15 minutes per day.

When dealing with deep learning models, there should always be a compromise between the use of a completely custom model based on a self-labelled dataset for more specific classifications and the used of already pretrained models for more general approaches to natural language processing.

Due to the lack of an already pretrained model, the following classifiers have been implemented from scratch using transfer learning on the base BERT model by Google:

- Gender prediction based on username and tweet content





- Hate speech targeted to women prediction based on Roberta Hugging Face transformer (phase II)

For the rest of the classifiers, a pretrained transformer model has been used in the inference phase:

- Hate speech prediction based on Hugging Face transformer (phase I)
- Sentiment analysis based on Hugging Face transformer
- Topic modelling based on Hugging Face model
- Translation Spanish-English and Tagalog-English based on a Hugging Face transformer.

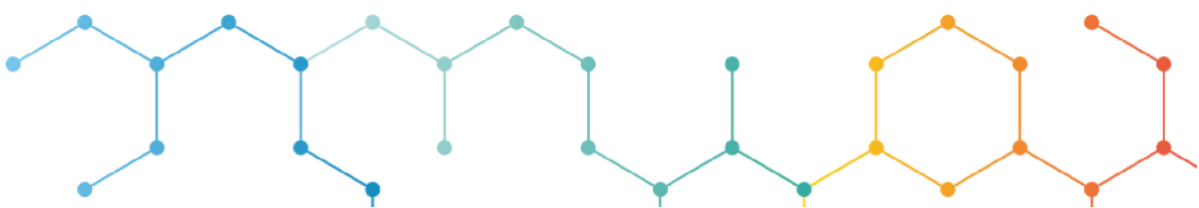
All the AI models used in the EWS project are based on a transformer architecture, which is a deep learning model based on an encoder-decoder architecture with an attention mechanism.

Transformers were originally design as language models for translation, so their functionality is better understood using a translation task as an example. They are trained with large amounts of data in a self-supervised way, hence no need for input labelled data is required and their main goal is to develop a statistical understanding of the language they are trained for, but do not implement well in any specific NLP task.

They require high volumes of computational resources so there are just a few companies in the world that can create meaningful models (OpenAI, HuggingFace, Alphabet, Meta, etc) and its carbon footprint is not at all neglectable. Due to these reasons, the best practice is always to use an out-of-the-box pre-trained model for a specific classification problem or do some transfer learning (fine-tuning on a specific task) on the pre-trained large language models available:

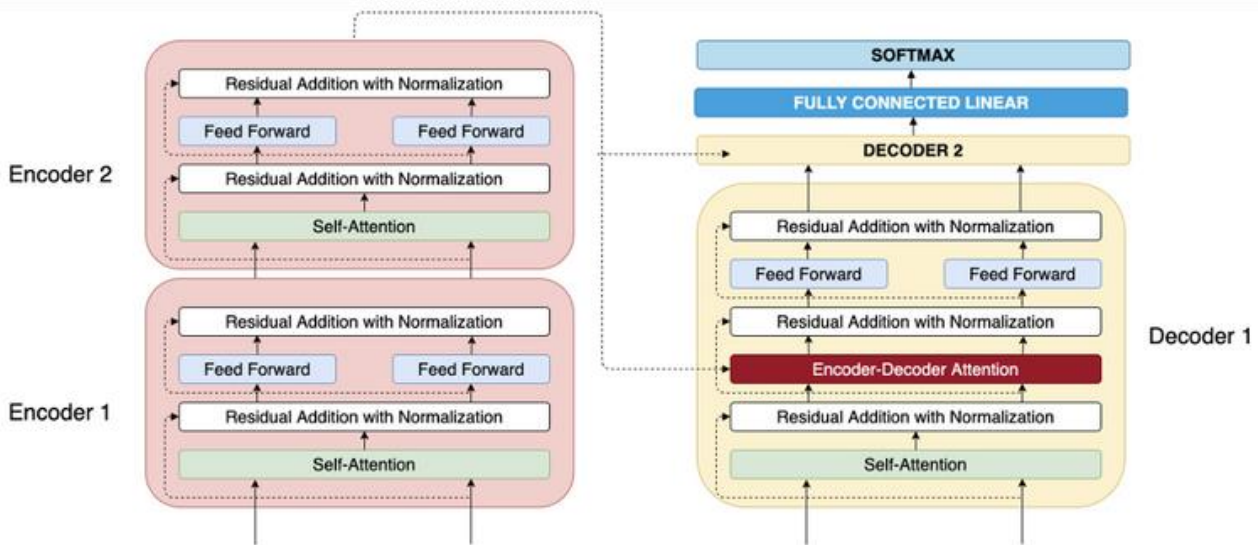
- GPT: the first pre-trained Transformer model, used for fine-tuning on various NLP tasks and obtained state-of-the-art results
- BERT: another large pre-trained model, this one designed to produce better summaries of sentences
- DistilBERT: A distilled version of BERT that is 60% faster, 40% lighter in memory, and still retains 97% of BERT's performance
- BART: large pre-trained model using the same architecture as the original Transformer model
- GPT-3: an even bigger version of GPT-2 that is able to perform well on a variety of tasks without the need for fine-tuning (called zero-shot learning)

Using a transfer learning mechanism to train our models always results in a reduction in the amount of time and resources needed, less labelled data to get high accuracy models and less environmental impact.





From the technical point of view, the transformer architecture is primarily composed of two blocks, one encoder and one decoder:



The encoder receives input and builds a representation of it (its features). This means that the model is optimized to acquire understanding from the input. Then the decoder uses the encoder's representation (features) along with other inputs to generate a target sequence. This means that the model is optimized for generating outputs.

In an English-Spanish translation model example, the encoder will capture the grammar and semantics of each English word presented in the corpus and represent them using vectors that will be sent to the decoder. The decoder will then translate those features from English to Spanish.

A key feature of Transformer models is that they are built with special layers called attention layers. This layer will tell the model to pay specific attention to certain words in the sentence you passed it when dealing with the representation of each word.

Going back to our previous example, given the input "You like this manual", a translation model will need to also attend to the adjacent word "You" to get the proper translation for the word "like", because in Spanish the verb "like" is conjugated differently depending on the subject. The rest of the sentence, however, is not useful for the translation of that word. In the same vein, when translating "this" the model will also need to pay attention to the word "manual", because "this" translates differently depending on whether the associated noun is masculine or feminine. Again, the other words in the sentence will not matter for the translation of "this". With more complex sentences (and more complex grammar rules), the model would need to pay special attention to words that might appear farther away in the sentence to properly translate each word.





1.2 Gender classification

Since Twitter does not request its users to provide their gender, there is no way to get that information using the Twitter API or any other scrapping tool. There have been many studies that try to address gender prediction such as the works of Burger, J. D et al [5], Ludu P. S. [6] or Verhoeven, B., Daelemans, W., & Plank, B. [7]. Different studies have conducted different approximations for the classification problem, depending on the input parameters for the model:

- Tweet content
- Username
- Profile description
- Profile picture
- User network

There are not publicly available state-of-the-art pretrained models for this classification problem, but there is a dataset containing Twitter information extracted from the Twitter API that has been manually labelled and it has been released to the general public by CrowdFlower AI. In the EWS project, this dataset has been used to train a new model based on BERT to predict the gender of the user writing each tweet. To make the prediction more robust, de dataset X, which is a compendium of the most relevant names in English for men and women has been used in the case of Uganda.

It shall be noted that since the CrowdFlower AI dataset focuses just on English tweets, the gender prediction model for Spanish and Tagalog has been made using the same dataset with the translated tweets. The translation has been performed using deep learning models as well, specifically.

Language	Translation model
Spanish	Helsinki-NLP/opus-mt-en-es ⁹
Tagalog	Helsinki-NLP/opus-mt-en-tl ¹⁰

1.2.1 Dataset

The contributors of the selected dataset were asked to simply view a Twitter profile and judge whether the user was a male, a female, or a brand (non-individual). The dataset contains 20,000 rows, each with a username, a random tweet, account profile and image, location, and even link and sidebar color.





The data was provided by the Data For Everyone Library on Crowdfunder[3], released under CC0: Public Domain license. It contains the following columns:

- unitid: a unique id for user
- _golden: whether the user was included in the gold standard for the model; TRUE or FALSE
- unitstate: state of the observation; one of finalized (for contributor-judged) or golden (for gold standard observations)
- trustedjudgments: number of trusted judgments (int); always 3 for non-golden, and what may be a unique id for gold standard observations
- lastjudgment_at: date and time of last contributor judgment; blank for gold standard observations
- gender: one of male, female, or brand (for non-human profiles)
- gender:confidence: a float representing confidence in the provided gender
- profile_yn: "no" here seems to mean that the profile was meant to be part of the dataset but was not available when contributors went to judge it

[1] <https://huggingface.co/Helsinki-NLP/opus-mt-en-es>

[2] <https://huggingface.co/Helsinki-NLP/opus-mt-en-tl>

[3] <https://www.kaggle.com/datasets/crowdfunder/twitter-user-gender-classification>

- profile_yn:confidence: confidence in the existence/non-existence of the profile
- created: date and time when the profile was created
- description: the user's profile description
- fav_number: number of tweets the user has favorited ·gender_gold: if the profile is golden, what is the gender?
- link_color: the link color on the profile, as a hex value
- name: the user's name
- profileyngold: whether the profile y/n value is golden
- profileimage: a link to the profile image
- retweet_count: number of times the user has retweeted (or possibly, been retweeted)
- sidebar_color: color of the profile sidebar, as a hex value
- text: text of a random one of the user's tweets
- tweet_coord: if the user has location turned on, the coordinates as a string with the format "[latitude, longitude]"
- tweet_count: number of tweets that the user has posted
- tweet_created: when the random tweet (in the text column) was created
- tweet_id: the tweet id of the random tweet
- tweet_location: location of the tweet; seems to not be particularly normalized ·user_timezone: the timezone of the user

The dataset has been used in several academic studies, as the one conducted by Qudar, M. M. A., & Mago, V[8] or the analysis on demographics carried out by Yildiz, D., Munson, J., Vitali, A., Tinati, R., & Holland, J. A [9].





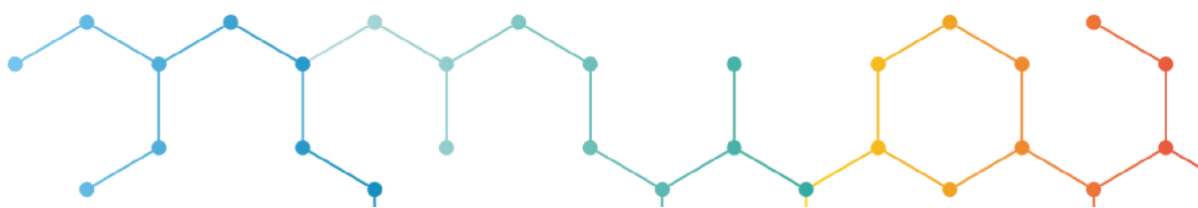
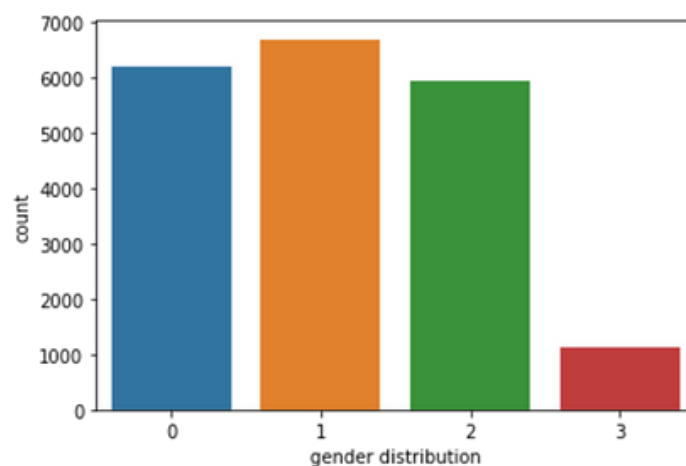
Before any model processing, a data preprocessing step shall be made. This preprocessing shall include data clearance and basic analysis of the information stored in the data set. First, the tweets should be cleaned up from hashtags, URLs, punctuation marks, etc. Then, the missing values should be removed from the sample:

```
def clean_tweet(tweet):
    stopwords = ["for", "on", "an", "a", "of", "and", "in", "the", "to", "from", "girl",
    "girls", "women", "woman", "s", "u", "t", "womens", "amp", "im", "m", "re"]

    if type(tweet) == np.float:
        return ""
    temp = tweet.lower()
    temp = re.sub(" ", "", temp) # to avoid
    removing contractions in english
    temp = re.sub("@[A-Za-z0-9_]+", "", temp)
    temp = re.sub("#[A-Za-z0-9_]+", "", temp)
    temp = re.sub(r'http\S+', "", temp)
    temp = re.sub("([!]?)", "", temp)
    temp = re.sub("[\.\*\?]", "", temp)
    temp = re.sub("[^a-z0-9]", "", temp)
```

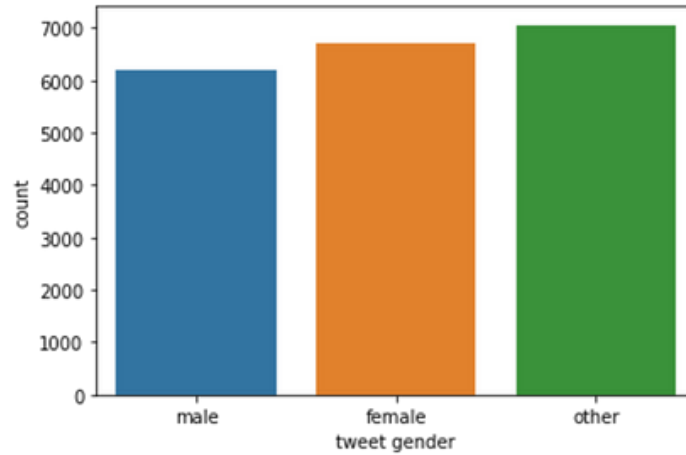
```
temp = temp.split()
temp = [w for w in temp if not w in stopwords]
temp = [w for w in temp if not w in STOPWORDS]
temp = " ".join(word for word in temp)
return temp
```

Now it shall be checked whether the classes are balanced. The following diagram shows the tweet distribution by gender, where zero represents males, one represents female, two represents brands and three represents unknown.





There is an imbalance between the 'Unknown' class and the rest of the classes, let's merge classes 2 and 3 into a single class.



Now that all the tweets are cleaned and the classes are balanced, we can proceed to tokenize the data, i.e. split the sentence into smaller chunks (words),

1.2.2 Pretrained base model

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google. In 2019, Google announced that it had begun leveraging BERT in its search engine, and by late 2020 it was using BERT in almost every English-language query. A 2020 literature survey concluded that "in a little over a year, BERT has become a ubiquitous baseline in NLP experiments", counting over 150 research publications analyzing and improving the model.

BERT was pretrained on two tasks: language modelling (15% of tokens were masked and BERT was trained to predict them from context) and next sentence prediction (BERT was trained to predict if a chosen next sentence was probable or not given the first sentence). As a result of the training process, BERT learns contextual embeddings for words. After pretraining, which is computationally expensive, BERT can be finetuned with less resources on smaller datasets to optimize its performance on specific tasks, in this case, gender prediction on twitter data.

1.2.2.1 Installation

To install a BERT base model for transfer learning, the following commands have to be introduced into the command prompt or inside a python Notebook:





```
pip install -qq
transformers
from transformers import BertTokenizer, BertForMaskedLM, BertModel    from
transformers.optimization import AdamW,get_linear_schedule_with_warmup
```

1.2.2.2 Initialization

The first step of the initialization is to load the model, as in the following command:

```
PRE_TRAINED_MODEL_NAME = 'bert-base-cased' bert_model =
BertModel.from_pretrained(PRE_TRAINED_MODEL_NAME)
```

Then, we shall proceed to convert the cleaned tweets into a numerical format understood by the BERT architecture. Here are the requirements:

- Add special tokens to separate sentences and do classification
- Pass sequences of constant length (introduce padding)
- Create an array of 0s (pad token) and 1s (real token) called attention mask

The Transformer BERT model also includes prebuild tokenizers that can transform the raw text into tensors suitable for the proposed architecture. To use those tokenizers, the following command shall be called:

```
tokenizer = BertTokenizer.from_pretrained(PRE_TRAINED_MODEL_NAME)
```

Although the input text will be tokenized after this step, it is a good practice to include special tokens for consistency:

- End of sentence: `tokenizer.cls_token`, `tokenizer.cls_token_id`
- Padding: `tokenizer.pad_token`, `tokenizer.pad_token_id`
- Unknown tokens (those not presented in the training set): `tokenizer.unk_token`, `tokenizer.unk_token_id`

Finally, a call to the function tokenizing the whole text shall be implemented:

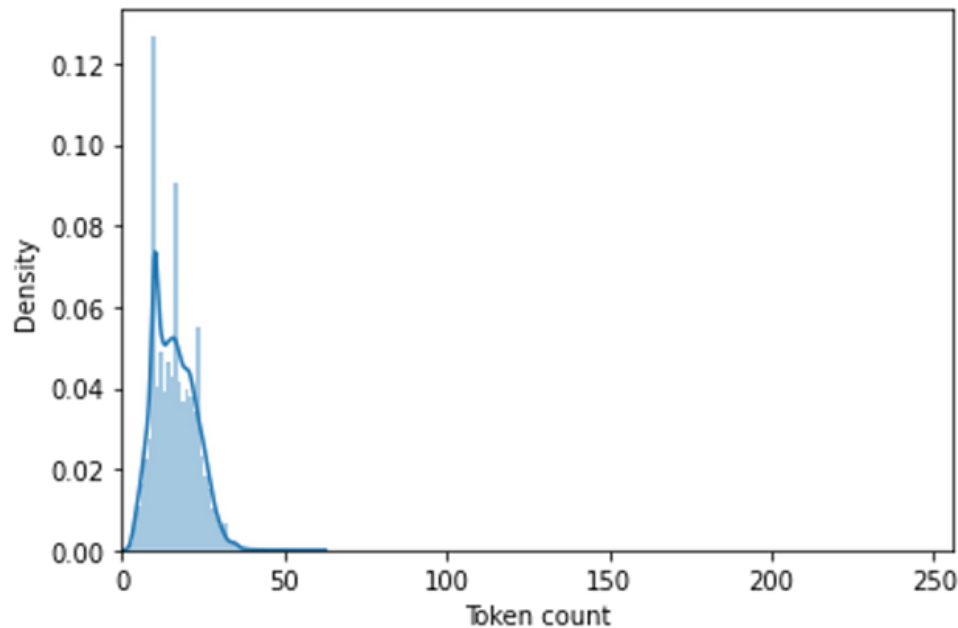
```
encoding = tokenizer.encode_plus( sample_txt,
max_length=32, add_special_tokens=True, # Add
'[CLS]' and '[SEP]' return_token_type_ids=False,
pad_to_max_length=True,
return_attention_mask=True, return_tensors='pt', #
Return PyTorch tensors
)
```

```
encoding.keys()
```





BERT works with fixed-length sequences. To choose the maximum sequence length, we plot the dataset distribution to check what is the average length of tokens in a tweet.



Most of the tweets seem to contain less than 35 tokens, so a maximum length of 50 will be chosen for safety reasons.

Finally, the loaders for a pytorch implementation will be designed. Pytorch has been selected as our preferred method to train the model because of its speed and robustness.

```
def create_data_loader(df, tokenizer, max_len, batch_size): ds
= GPGenderDataset( tweets=df.cleaned_text.to_numpy(),
targets=df.gender.to_numpy(), tokenizer=tokenizer,
max_len=max_len
)

return DataLoader(
ds,
batch_size=batch_size, num_workers=4
)
```

Finally, the architecture of the neural network shall be designed and initialized. Our classifier delegates most of the heavy lifting to the BertModel. We use a dropout layer for some regularization and a fully connected layer for our output. Note that we're returning the raw output of the last layer since that is required for the cross-entropy loss function in PyTorch to work.





1.2.2.3 Training & Evaluation

For the EWS project, two different models have been trained and evaluated. The first model takes the tweet content as input and predicts the gender of the user based on those inputs. The second model takes the username as input and predicts the gender based on the name.

1.2.2.3.1 Tweets as input variables

To reproduce the training procedure from the BERT paper[9], we'll use the AdamW optimizer provided by Hugging Face. It corrects weight decay, so it's similar to the original paper.

After eleven epochs using the advance-analysis cluster defined in 1.4.1. we get the following results in our training and validation sets for Uganda:

Split	Loss	Accuracy
Training	0.12846622336601432	0.9861730801358802
Validation	0.981700923291464	0.9200400801603206

After eleven epochs using the advance-analysis cluster defined in 1.4.1. we get the following results in our training and validation sets for Colombia:

After eleven epochs using the advance-analysis cluster defined in 1.4.1. we get the following results in our training and validation sets for Philipinnes:

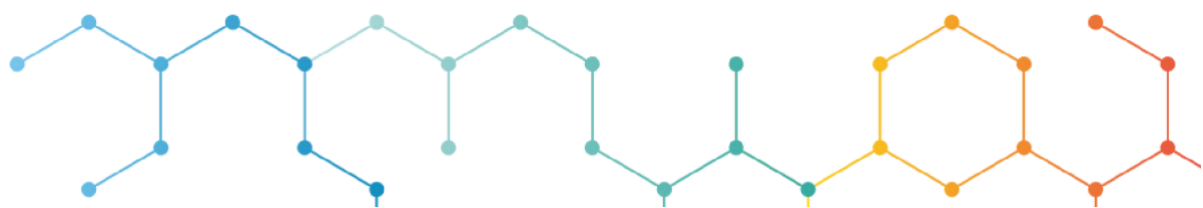
Split	Loss	Accuracy
Training	0.534729087892481	0.91267304u4i32022
Validation	1.352675435890876	0.8687965656789754

1.2.2.3.2 Username as input variables

To reproduce the training procedure from the BERT paper[9], we'll use the AdamW optimizer provided by Hugging Face. It corrects weight decay, so it's similar to the original paper.

After eleven epochs using the advance-analysis cluster defined in 1.4.1. we get the following results in our training and validation sets for Uganda:

Split	Loss	Accuracy
Training	0.618293478970843	0.8976536826878902
Validation	1.382673943087086	0.783584982323t409





After eleven epochs using the advance-analysis cluster defined in 1.4.1. we get the following results in our training and validation sets for Colombia:

Split	Loss	Accuracy
Training	0.100078978499902	0.9865378294782011
Validation	0.368740299782301	0.9578382974014938

After eleven epochs using the advance-analysis cluster defined in 1.4.1. we get the following results in our training and validation sets for Philipinnes:

Split	Loss	Accuracy
Training	6.012075960483092	0.4576382749392738
Validation	6.578493729204857	0.4133884895937289

1.3 Hate speech classification

Hate speech detection is the task of detecting if communication such as text, audio, and so on contains hatred and or encourages violence towards a person or a group of people. This is usually based on prejudice against 'protected characteristics' such as their ethnicity, gender, sexual orientation, religion or age.

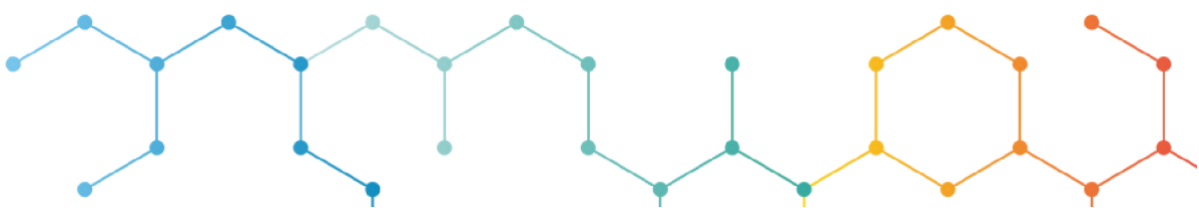
For the EWS project hate speech classification shall be made in three different languages: English, Spanish and Tagalog.

As a first approach, the hatespeech analysis was made using pretrained models already available on huggingface. Nevertheless, this approach had some limitations when trying to capture just hatespeech targeted to women.

During a second phase, the hatespeech analysis has been performed using custom trained algorithms based on three different manually labelled datasets.

1.3.1. Phase I

Hate speech classification in English has been made using bert_sequence_classifier_dehatebert_mono model [10]. This model has been selected because it is already available as a spark NLP migrated model and its enhanced performance on large datasets make it really suitable for EWS project.





The `bert_sequence_classifier_dehatebert_mono` model has been trained finetuning the multilingual BERT model but just on English language. The model is trained[1] with different learning rates and the best validation score achieved is 0.726030 for a learning rate of $2e5$.

It shall be noted that this model has been trained to detect general hate speech discussion and it should not be used as a starting point for transfer learning on more specific hate speech topics. The multilingual BERT model should be used instead.

The data source code for the classifier can be found at https://nlp.johnsnowlabs.com/2021/11/03/bert_sequence_classifier_dehatebert_mono_en.html.

Hate speech classification in Spanish could use `pysentimiento/bertweet-hate-speech` hugging face model, nevertheless, this model is not optimized for running on a distributed environment, i.e. every tweet classification shall be made tweet by tweet on a single machine, without parallelization. Trying to analyzed a dataset of N tweets in Spanish would require $N \cdot 60$ seconds to complete. Spark NLP can mitigate this issue by transferring the model to a spark NLP schema, but the model should be migrated to the Spark NLP platform in this case.

This approach has been tried several times without success because the base BERT model use for transfer learning is unknown and none of the BERT models hosted in Spark NLP platform seem to inherit all the properties defined in the Spanish hate speech model.

Due to this limitation, it has been decided that the best approach for now should be to work directly in English by translating the Spanish tweets to English using another translation transformer (`translate_mul_en[2]`).

Hate speech classification in Tagalog does not have any publicly available models at the moment this memory was written. There is the `hate_speech_filipino` dataset on hugging face[1]¹ so the first approach for the EWS was to implement custom hate speech model based on the multilingual BERT (it shall be noted that no pretrained embeddings have been found in Tagalog either). The accuracy achieved during the training & evaluation phase of the modellization was so poor ($\sim 39\%$), that translating from talaglog to English seemed the best workaround. The notebook containing the source code for the failed prototype model can be found at the zip file included with this memory.

¹ [1] Training code can be found here <https://github.com/punyajoy/DE-LIMIT> [2] https://nlp.johnsnowlabs.com/2021/01/03/translate_mul_en_xx.html





1.3.1.1

Translators

1.3.1.1.1 Spanish-English

Marian is an efficient, free Neural Machine Translation framework written in pure C++ with minimal dependencies. It is mainly being developed by the Microsoft Translator team. Many academic (most notably the University of Edinburgh and in the past the Adam Mickiewicz University in Poznań) and commercial contributors help with its development.

It is currently the engine behind the Microsoft Translator Neural Machine Translation services and being deployed by many companies, organizations and research projects (see below for an incomplete list).

Note that this is a very computationally expensive module especially on larger sequence. The use of an accelerator such as GPU is recommended.

1.3.1.1.1.1 Model description

Model Name:	translate_mul_en
Type:	pipeline
Compatibility:	Spark NLP 2.7.0+
Edition:	Official

[1] https://huggingface.co/datasets/hate_speech_filipino

Language:	xx
------------------	-----------

1.3.1.1.1.2 Implementation

```
pipeline = PretrainedPipeline("translate_mul_en", lang = "xx")
annotation = pipeline.annotate(sparkDF_Colombia,column="text")
```

```
res_Colombia =
annotation.select(annotation.date,annotation.nretweets,annotation.Country,annotation.Top
ic,annotation.gender,annotation.HateSpeech,F.explode(F.arrays_zip(annotation.sentence.r
esult,
annotation.translation.result)).alias("col"))\
```





```
.select(annotation.date,F.expr("col['0']").alias("tweet"),F.expr("col['1']").alias("text"),annotation.nretweets,annotation.Country,annotation.Topic,annotation.gender,annotation.HateSpeech)
```

1.3.1.1.2 Tagalog-English

Marian framework, as in 1.3.1.2 section, has also being used to translate from Tagalog to English.

1.3.1.1.2.1 Model description

Model Name:	translate_tl_en
Type:	pipeline
Compatibility:	Spark NLP 2.7.0+
Edition:	Official
Language:	xx

1.3.1.1.2.2 Implementation

```
pipeline = PretrainedPipeline("translate_tl_en", lang = "xx") annotation =  
pipeline.annotate(sparkDF_Philippines,column="text") res_Philippines  =  
annotation.select(annotation.date,annotation.nretweets,annotation.Country,annotation.Top  
ic,annotation.gender,annotation.HateSpeech,F.explode(F.arrays_zip(annotation.sentence.r esult,  
annotation.translation.result)).alias("col"))\  
.select(annotation.date,F.expr("col['0']").alias("tweet"),F.expr("col['1']").alias("text"),annotation.  
nretweets,annotation.Country,annotation.Topic,annotation.gender,annotation.HateSpeech)
```





1.3.1.2 Hate
speech

1.3.1.2.1 Model description

Model Name:	bert_sequence_classifier_dehatebert_mono
Compatibility:	Spark NLP 3.3.2+
License:	Open Source
Edition:	Official
Input Labels:	[token, document]
Output Labels:	[class]
Language:	en
Case sensitive:	false
Max sentence length:	512

1.3.1.2.2 Implementation

```
document_assembler = DocumentAssembler() \  
  .setInputCol('text') \  
  .setOutputCol('document') tokenizer  
  
= Tokenizer() \  
  
.setInputCols(['document']) \  
  
.setOutputCol('token')  
  
sequenceClassifier = BertForSequenceClassification \  

```





```
.pretrained('bert_sequence_classifier_dehatebert_mono', 'en') \
    .setInputCols(['token', 'document']) \
    .setOutputCol('class') \
.setCaseSensitive(True) pipeline =
Pipeline(stages=[
    document_assembler,
tokenizer,  sequenceClassifier
])
```

1.3.1.2.3 Examples

Here is a list of examples to showcase how information is classified as hate or non-hate:

Sentence	Hate percentage
I hate this school	0.013
He is just a traitor	0.01
This is bullshit but it's not hate	0.022
This is hate because immigrants must be annihilated	0.835
What a cheap slut is XXXX!	0.97
Abortion should not be illegal	0.039
Abortion should be illegal	0.68

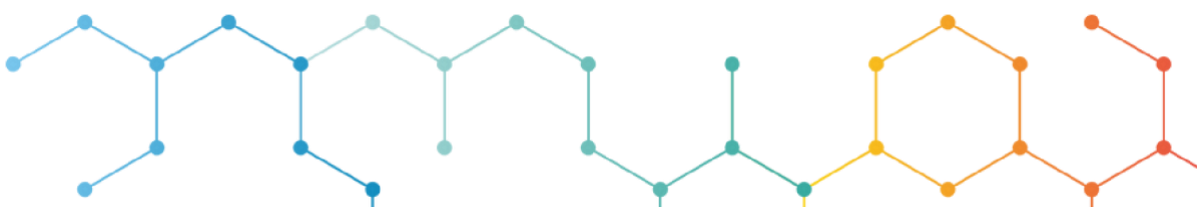
1.3.2. Phase II

The input datasets manually labelled by UNDP volunteers for each country has been used to train the different models introduced in this section. These datasets can be found at the annexed files of this manual.

1.3.2.1 Monolingual models

Monolingual hate speech classifiers are trained using different algorithms to select the one outperforming the task:

- Logistic regression
- LSTM neural network
- Transformer fine tuned using DistilBERT





1.3.2.1.1 Logistic regression

This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure.





1.3.2.1.1.1. Implementation scikit-learn

```
x_train, x_test, y_train, y_test = train_test_split(df['text'], df['label'], test_size=0.2,
random_state=1)

count_vectorizer = feature_extraction.text.CountVectorizer() # Using the common
bag of words technique
train_vectors = count_vectorizer.fit_transform(x_train)
baseline_model = linear_model.LogisticRegression()
f1_scores = model_selection.cross_val_score(baseline_model, train_vectors,
y_train, cv=3, scoring="f1")
accuracy_scores = model_selection.cross_val_score(baseline_model,
train_vectors, y_train, cv=3, scoring="accuracy")
print(f"Cross Validation Accuracy Scores:", accuracy_scores)
print(f"Cross Validation Accuracy f1_score:", f1_scores)

baseline_model.fit(train_vectors, y_train)

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)

test_vectors = count_vectorizer.transform(x_test)
baseline_predict_test = baseline_model.predict(test_vectors)
print("Accuracy:", accuracy_score(baseline_predict_test, y_test))
print("F1_score:", f1_score(baseline_predict_test, y_test))

def plot_confusion_matrix(y_preds, y_true, labels):
    cm = confusion_matrix(y_true, y_preds, normalize="true")
    fig, ax = plt.subplots(figsize=(6, 6))
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
    disp.plot(cmap="Blues", values_format=".2f", ax=ax, colorbar=False)
```

1.3.2.1.1.2. Results

In the three different monolingual models, the logistic regression algorithm was always the faster to train but the performance was the lowest in all cases and it was highly sensitive to imbalance datasets.

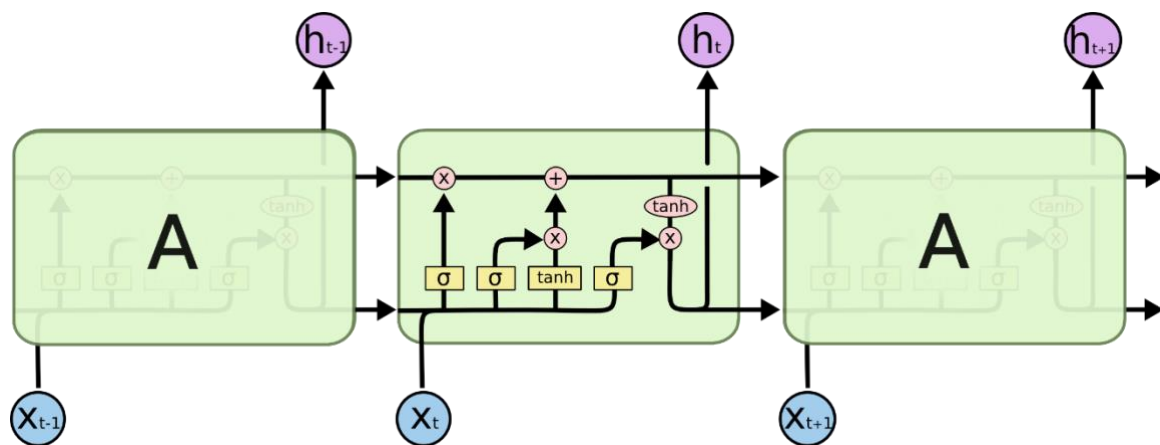




1.3.2.1.2. LSTM neural networks

A more complex alternative to a traditional machine learning approach based on logistic regression is neural networks, which usually improves the performance but takes more time and effort to implement.

Classic classification vanilla neural networks suffer from short-term memory. Also, a big drawback is the vanishing gradient problem, i.e. during backpropagation the gradient becomes so small that it tends to 0 and such a neuron is of no use in further processing. To overcome these inconvenient, LSTMs (Long Short Term Memory) neural networks, which are a special kind of recurrent neural network, efficiently improve performance by memorizing the relevant information that is important inside a sentence and finds hidden patterns.



The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is the predefined path where information just flows along the different layers with minor changes.

The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer."

The next step is to decide what new information we're going to store in the cell state. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a tanh layer creates a vector of new candidate values that could be added to the state. In the next step, we'll combine these two to create an update to the state.

Finally, we need to decide what we're going to output. This output will be based on our cell state but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh (to push the values to be between -1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.





1.3.2.1.2.1. Implementation on tensorflow

```
def LSTMTrainer(df):
    X_train, X_test, y_train, y_test = train_test_split(df["text"], df["label"],
test_size=0.2, random_state=1)
    X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.25,
random_state=1)

    vocabulary_size = 5000

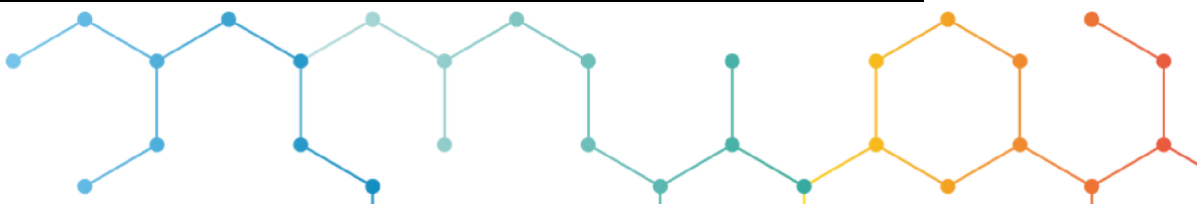
    # Tweets have already been preprocessed hence dummy function will be passed in
    # to preprocessor & tokenizer step
    count_vector = CountVectorizer(max_features=vocabulary_size,
#                                     ngram_range=(1,2), # unigram and bigram
preprocessor=lambda x: x,
tokenizer=lambda x: x)
    #tfidf_vector = TfidfVectorizer(lowercase=True, stop_words='english')


    # Fit the training data
    X_train = count_vector.fit_transform(X_train).toarray()

    # Transform testing data
    X_test = count_vector.transform(X_test).toarray()

    max_words = 5000
    max_len=50

    def tokenize_pad_sequences(text):
        """
        This function tokenize the input text into sequences of intergers and then
        pad each sequence to the same length
        """
        # Text tokenization
        tokenizer = Tokenizer(num_words=max_words, lower=True, split=' ')
        tokenizer.fit_on_texts(text)
        # Transforms text to a sequence of integers
        X = tokenizer.texts_to_sequences(text)
        # Pad sequences to the same length
        X = pad_sequences(X, padding='post', maxlen=max_len)
        # return sequences
        return X, tokenizer
```





```
X, tokenizer = tokenize_pad_sequences(df['text'])

y = pd.get_dummies(df['label'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=1)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.25,
random_state=1)

def f1_score(precision, recall):
    ''' Function to calculate f1 score '''

    f1_val = 2*(precision*recall)/(precision+recall+K.epsilon())
    return f1_val

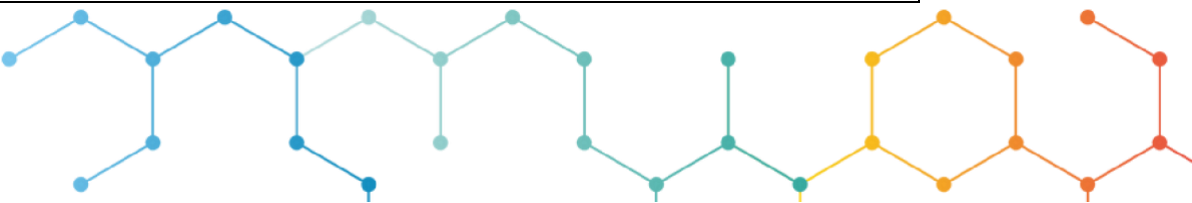
vocab_size = 5000
embedding_size = 32
epochs=20
learning_rate = 0.1
decay_rate = learning_rate / epochs
momentum = 0.8

sgd = SGD(lr=learning_rate, momentum=momentum, decay=decay_rate,
nesterov=False)
# Build model
model= Sequential()
model.add(Embedding(vocab_size, embedding_size, input_length=max_len))
model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Bidirectional(LSTM(32)))
model.add(Dropout(0.4))
model.add(Dense(2, activation='softmax'))

X, tokenizer = tokenize_pad_sequences(df['text'])

print(model.summary())

# Compile model
model.compile(loss='categorical_crossentropy', optimizer=sgd,
metrics=['accuracy', Precision(), Recall()])
```





```
# Train model
```

```
batch_size = 64
```

```
history = model.fit(X_train, y_train,
                    validation_data=(X_val, y_val),
                    batch_size=batch_size, epochs=epochs, verbose=1)
```

```
# Evaluate model on the test set
```

```
loss, accuracy, precision, recall = model.evaluate(X_test, y_test, verbose=0)
```

```
# Print metrics
```

```
print("")
```

```
print('Accuracy : {:.4f}'.format(accuracy))
```

```
print('Precision : {:.4f}'.format(precision))
```

```
print('Recall : {:.4f}'.format(recall))
```

```
print('F1 Score : {:.4f}'.format(f1_score(precision, recall)))
```

```
def plot_confusion_matrix(model, X_test, y_test):
```

```
    """Function to plot confusion matrix for the passed model and the data"""
```

```
    sentiment_classes = ['No', 'Yes']
```

```
    # use model to do the prediction
```

```
    y_pred = model.predict(X_test)
```

```
    # compute confusion matrix
```

```
    cm = confusion_matrix(np.argmax(np.array(y_test),axis=1), np.argmax(y_pred,
axis=1), normalize="true")
```

```
    # plot confusion matrix
```

```
    fig, ax = plt.subplots(figsize=(6, 6))
```

```
    disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=sentiment_classes)
```

```
    disp.plot(cmap="Blues", values_format=".2f", ax=ax, colorbar=False)
```

```
    plt.title("Normalized confusion matrix")
```

```
    plt.show()
```

```
plot_confusion_matrix(model, X_test, y_test)
```

1.3.2.1.2.2. Results

In the three different monolingual models, the LSTM network results were always as good or even slightly better than the other algorithms. The training phase took longer than the traditional logic





regression but was 10 times faster than using a fine-tune BERT model for the hate speech target to women classification.

Below you can find the results comparing the three best performance algorithms on the three different languages, with and without class imbalance.

Imbalance	Language	Model	Accuracy	F1-score	Training time
Yes	English	LR	0.8733	0.2875	1.28s
Yes	English	LSTM	0.7729	0.4381	34.82s
Yes	English	BERT	0.7286	0.4188	2.57min
Yes	Spanish	LR	0.7698	0.6710	1.09s
Yes	Spanish	LSTM	0.8667	0.4611	27.77s
Yes	Spanish	BERT	0.8826	0.6998	3.39min
Yes	Tagalog	LR	0.9832	0.2222	0.93s
Yes	Tagalog	LSTM	0.9808	0.2381	40.05s
Yes	Tagalog	BERT	1	0.1245	7min

Imbalance	Language	Model	Accuracy	F1-score	Training time
No	English	LSTM	0.9607	0.9514	31.38s
No	English	BERT	0.96	0.95001	4.16min
No	Spanish	LSTM	0.7729	0.6245	38.19s
No	Spanish	BERT	0.9621	0.9611	4.93min
No	Tagalog	LSTM	0.9574	0.9327	48.74s
No	Tagalog	BERT	0.9890	0.9187	8.12min

Again, it shall be noted that the LSTM neural network is really affected by an imbalance dataset.

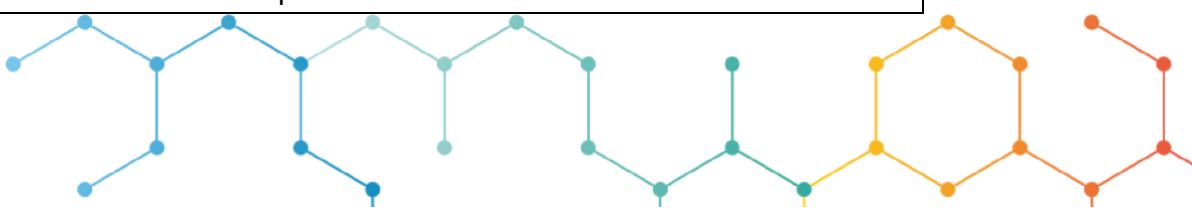
1.3.2.1.3. Fine-tune BERT model

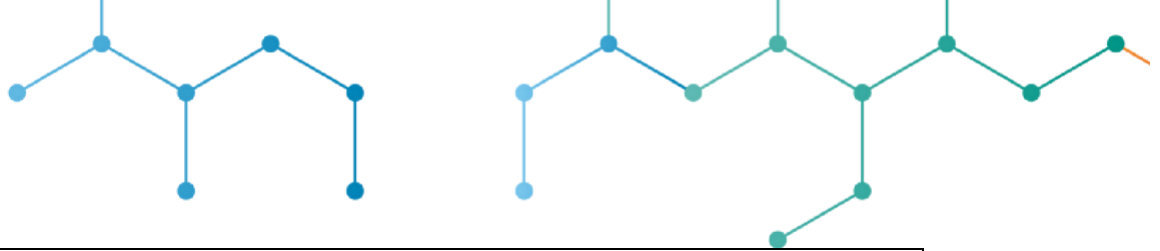
BERT is a deep learning model that has given state-of-the-art results on a wide variety of natural language processing tasks. It stands for Bidirectional Encoder Representations for Transformers. It has been pre-trained on Wikipedia and BooksCorpus and requires task-specific fine-tuning.

1.3.2.1.3.1. Implementation on HuggingFace library

Huggingface Datasets is a library for easily accessing and sharing datasets for Natural Language Processing tasks. These tool uses Huggingface powerful data processing methods to quickly get your dataset ready for training in a deep learning model. Backed by the Apache Arrow format, it processes large datasets with zero-copy reads without any memory constraints for optimal speed and efficiency.

```
def CreateHFDataset(path):
    #Hugging face datasets based on Apache Arrow
```





```
twitter = load_dataset("csv",data_files=path)
#twitter = load_dataset("csv",data_files=file_location)
dataset = twitter['train']
# 90% train, 10% test + validation
train_testvalid = dataset.train_test_split(test_size=0.1)
# Split the 10% test + valid in half test, half valid
test_valid = train_testvalid["test"].train_test_split(test_size=0.5)

# gather everyone if you want to have a single DatasetDict
datasets = DatasetDict({
    "train": train_testvalid["train"],
    "test": test_valid["test"],
    "validation": test_valid["train"]})

return datasets
```

There are several subword tokenization algorithms that are commonly used in NLP, one of them is WordPiece, which is used by the BERT and DistilBERT tokenizers.

Hugging face transformers provide an AutoTokenizer class that allows to quickly load the tokenizer associated with a pretrained model.

```
def SubWordTokenization(dataset):
    model_ckpt = "distilbert-base-uncased"
    tokenizer = AutoTokenizer.from_pretrained(model_ckpt)

    def tokenize(batch):
        return tokenizer(batch["text"],padding=True,truncation=True)


    hatespeech_encoded = dataset.map(tokenize,batched=True,batch_size=None)

    return hatespeech_encoded
```

It is nowadays common practice in NLP to use transfer learning to train a neural on one task, and then adapt it to or fine-tune it on a new task. This allows the transformer network to make use of the knowledge learned. We will follow this approach to take advantage of the language model already pretrained on BERT.

```
def FineTuneTransformer(tokenData):
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```





```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

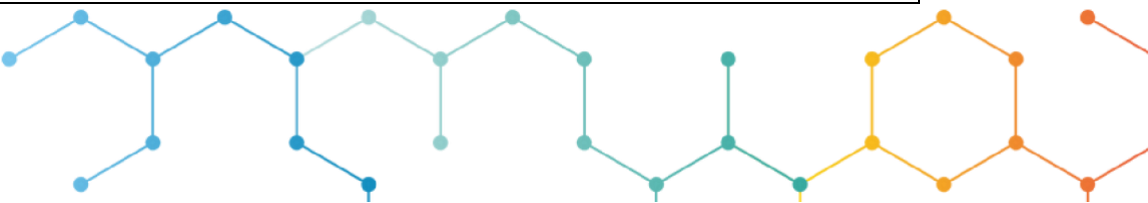
model_ckpt = "distilbert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_ckpt)
num_labels = 2

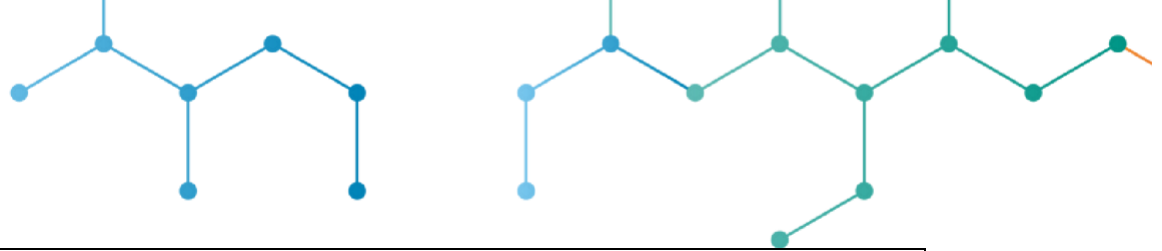
model = (AutoModelForSequenceClassification
        .from_pretrained(model_ckpt, num_labels=num_labels, problem_type =
"single_label_classification")
        .to(device))

batch_size = 64
logging_steps = len(tokenData["train"]) // batch_size
model_name = f"{model_ckpt}-finetuned-EWS-hatespeech"
training_args = TrainingArguments(output_dir=model_name,
                                  num_train_epochs=2,
                                  learning_rate=2e-5,
                                  per_device_train_batch_size=batch_size,
                                  per_device_eval_batch_size=batch_size,
                                  weight_decay=0.01,
                                  evaluation_strategy="epoch",
                                  disable_tqdm=False,
                                  logging_steps=logging_steps,
                                  push_to_hub=False,
                                  log_level="error")

def compute_metrics(pred):
    labels = pred.label_ids
    preds = pred.predictions.argmax(-1)
    f1 = f1_score(labels, preds, average="weighted")
    acc = accuracy_score(labels, preds)
    return {"accuracy": acc, "f1": f1}

trainer = Trainer(model=model, args=training_args,
                  compute_metrics=compute_metrics,
                  train_dataset=tokenData["train"],
                  eval_dataset=tokenData["validation"],
                  tokenizer=tokenizer)
trainer.train()
```





return trainer

During the model evaluation stage, performance evaluation metrics shall be provided to ensure that the model is working properly. For this, the input dataset has been split into train/validation and test data.

It shall be noted that to check that the model is working properly in such imbalanced input datasets, the test data has been retrieved prior to any imbalance treatment over the data.

The test data contains the correct labels (observed labels) for all data instances. These observed labels are used to compare with the predicted labels for performance evaluation after classification.

The confusion matrix is a two by two table that contains four outcomes produced by a binary classifier. Various measures, such as error-rate, accuracy, specificity, sensitivity, and precision, are derived from the confusion matrix. Moreover, several advanced measures, such as ROC and precision-recall, are based on them.

```
def EvaluateModel(trainer,tokenData):
    preds_output = trainer.predict(tokenData["validation"])
    print(preds_output.metrics)

    y_preds = np.argmax(preds_output.predictions, axis=1)
    y_valid = np.array(tokenData["validation"]["label"])
    labels = ["No","Yes"]

    def plot_confusion_matrix(y_preds, y_true, labels):
        cm = confusion_matrix(y_true, y_preds, normalize="true")
        fig, ax = plt.subplots(figsize=(6, 6))
        disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
        disp.plot(cmap="Blues", values_format=".2f", ax=ax, colorbar=False)
        plt.title("Normalized confusion matrix")
        plt.show()

    plot_confusion_matrix(y_preds, y_valid, labels)
```

1.3.2.2. Results

In the three different monolingual models, the BERT fine-tuning model results were really close to the performance achieved using LSTM networks. The training phase took 10 times more than any other model for the hate speech target to women classification so this should not be our preferred solution for monolingual modelling.





Below you can find the results comparing the three best performance algorithms on the three different languages, with and without class imbalance.

Imbalance	Language	Model	Accuracy	F1-score	Training time
Yes	English	LR	0.8733	0.2875	1.28s
Yes	English	LSTM	0.7729	0.4381	34.82s
Yes	English	BERT	0.7286	0.4188	2.57min
Yes	Spanish	LR	0.7698	0.6710	1.09s
Yes	Spanish	LSTM	0.8667	0.4611	27.77s
Yes	Spanish	BERT	0.8826	0.6998	3.39min
Yes	Tagalog	LR	0.9832	0.2222	0.93s
Yes	Tagalog	LSTM	0.9808	0.2381	40.05s
Yes	Tagalog	BERT	1	0.1245	7min

Imbalance	Language	Model	Accuracy	F1-score	Training time
No	English	LSTM	0.9607	0.9514	31.38s
No	English	BERT	0.96	0.95001	4.16min
No	Spanish	LSTM	0.7729	0.6245	38.19s
No	Spanish	BERT	0.9621	0.9611	4.93min
No	Tagalog	LSTM	0.9574	0.9327	48.74s
No	Tagalog	BERT	0.9890	0.9187	8.12min

Again, it shall be noted that the BERT fine-tuned model is really affected by an imbalance dataset.

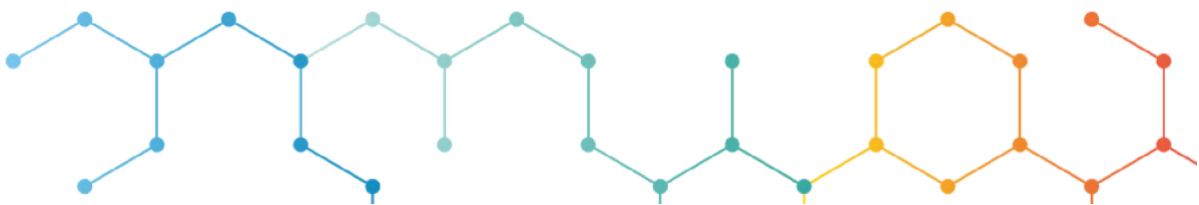
1.3.2.3. Multilingual model

There are several advantages of using a multilingual model instead of having multiple monolingual models at the same time:

- Easiness of maintenance
- By pretraining on huge corpora across many languages, these multilingual transformers enable zero-shot cross-lingual transfer. This means that a model that is fine-tuned on one language can be applied to others without any further training
- These models are well suited for “code-switching,” where a speaker alternates between two or more languages or dialects in the context of a single conversation. (e.g. English-Swahili in Uganda, Tagalog-English in Philippines)

Roberta models are the preferred solution for multilingual classification. These models use masked language modeling as a pretraining objective, but they are trained jointly on texts in over one hundred languages. By pretraining on huge corpora across many languages, these multilingual transformers enable zero-shot cross-lingual transfer.







```
def SentencePieceTokenization(dataset):
    model_ckpt = "xlm-roberta-base"
    tokenizer = AutoTokenizer.from_pretrained(model_ckpt)

    def tokenize(batch):
        return tokenizer(batch["text"],padding=True,truncation=True)

    hatespeech_encoded = dataset.map(tokenize,batched=True,batch_size=None)

    return hatespeech_encoded
```

It is nowadays common practice in NLP to use transfer learning to train a neural on one task, and then adapt it to or fine-tune it on a new task. This allows the transformer network to make use of the knowledge learned. We will follow this approach to take advantage of the language model already pretrained on multilingual Roberta.

```
def FineTuneTransformer(tokenData):
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

    model_ckpt = "xlm-roberta-base"
    tokenizer = AutoTokenizer.from_pretrained(model_ckpt)
    num_labels = 2

    model = (AutoModelForSequenceClassification
              .from_pretrained(model_ckpt, num_labels=num_labels,problem_type =
"single_label_classification")
              .to(device))

    batch_size = 32
    logging_steps = len(tokenData["train"]) // batch_size
    model_name = f"{model_ckpt}-finetuned-EWS-multihatespeech"
    training_args = TrainingArguments(output_dir=model_name,
                                     num_train_epochs=2,
                                     learning_rate=2e-5,
                                     per_device_train_batch_size=batch_size,
                                     per_device_eval_batch_size=batch_size,
                                     weight_decay=0.01,
                                     evaluation_strategy="epoch",
                                     disable_tqdm=False,
```





```
logging_steps=logging_steps,  
push_to_hub=False,  
log_level="error")  
  
def compute_metrics(pred):  
    labels = pred.label_ids  
    preds = pred.predictions.argmax(-1)  
    f1 = f1_score(labels, preds, average="weighted")  
    acc = accuracy_score(labels, preds)  
    return {"accuracy": acc, "f1": f1}  
  
trainer = Trainer(model=model, args=training_args,  
                  compute_metrics=compute_metrics,  
                  train_dataset=tokenData["train"],  
                  eval_dataset=tokenData["validation"],  
                  tokenizer=tokenizer)  
trainer.train()  
  
return trainer
```

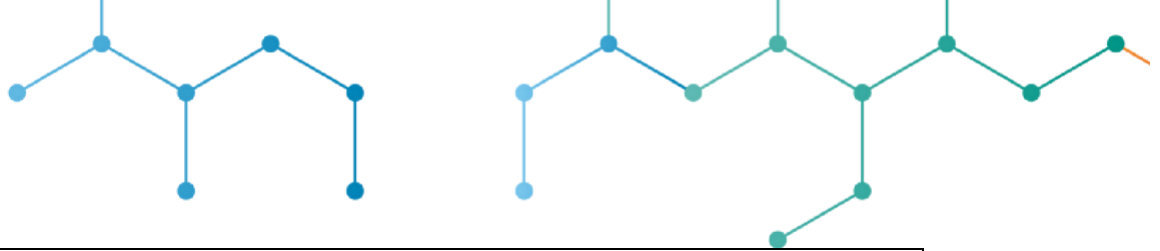
During the model evaluation stage, performance evaluation metrics shall be provided to ensure that the model is working properly. For this, the input dataset has been split into train/validation and test data. It shall be noted that to check that the model is working properly in such imbalanced input datasets, the test data has been retrieved prior to any imbalance treatment over the data.

The test data contains the correct labels (observed labels) for all data instances. These observed labels are used to compare with the predicted labels for performance evaluation after classification.

The confusion matrix is a two by two table that contains four outcomes produced by a binary classifier. Various measures, such as error-rate, accuracy, specificity, sensitivity, and precision, are derived from the confusion matrix. Moreover, several advanced measures, such as ROC and precision-recall, are based on them.

```
def EvaluateModel(trainer,tokenData):  
    preds_output = trainer.predict(tokenData["validation"])  
    print(preds_output.metrics)  
  
    y_preds = np.argmax(preds_output.predictions, axis=1)  
    y_valid = np.array(tokenData["validation"]["label"])  
    labels = ["No", "Yes"]
```





```
def plot_confusion_matrix(y_preds, y_true, labels):
    cm = confusion_matrix(y_true, y_preds, normalize="true")
    fig, ax = plt.subplots(figsize=(6, 6))
    disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
    disp.plot(cmap="Blues", values_format=".2f", ax=ax, colorbar=False)
    plt.title("Normalized confusion matrix")
    plt.show()

plot_confusion_matrix(y_preds, y_valid, labels)
```

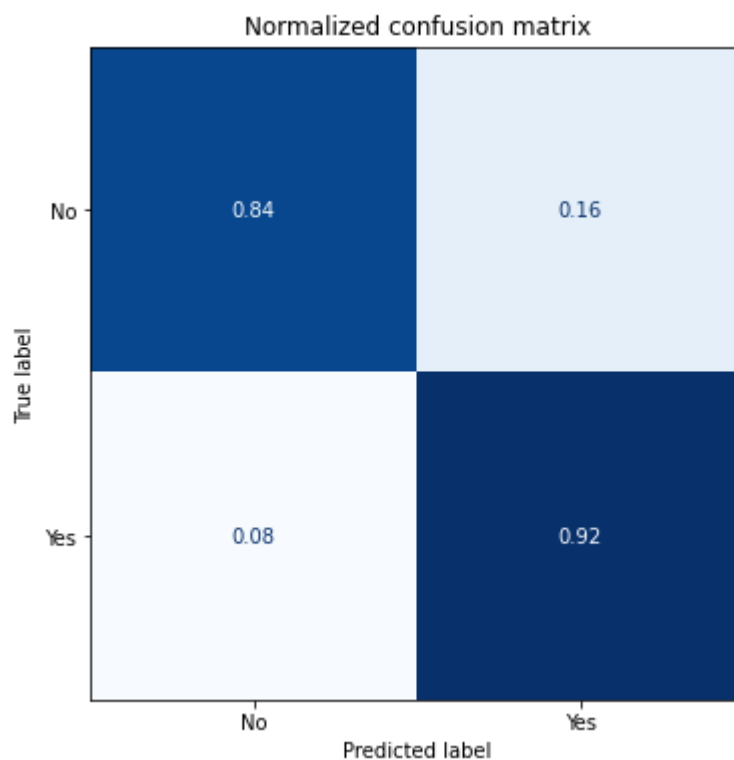
1.3.2.4. Results

Due to the great evidence of the affection of imbalance datasets to all models under analysis, in this case, it has been decided to preprocess the input dataset (which is a combination of the different datasets used in the monolingual models) to treat class imbalance.

These are the evaluation metrics for the final model using 1k-fold cross validation technique (k = 10) to decrease the bias of the metrics:

Accuracy	F1-score	Training time
0.91154	0.90254	146.75

And this is the confusion matrix for the final model:



Again, it shall be noted that the Roberta fine-tuned model is really affected by an imbalance dataset.





1.3.2.5. Hate speech model selection

As a summary, the best accuracy for the classification task is achieved using monolingual models trained independently for each language. Nevertheless, the multilingual model approach had an accuracy good enough to select it as a viable alternative with much more flexibility, lower maintenance, and easiness for future country extensions, so it has been agreed with the gender team to include the multilingual model as our preferred classifier.

The deployment of the model has been included in appendix B, please refer to “phase II” section for further analysis on the source code and additional information on memory constraints. The input datasets manually labelled by UNDP volunteers for each country has been used to train the different models introduced in this section. These datasets can be found at the annexed files of this manual.

1.4 Sentiment analysis

Sentiment analysis looks at the emotion expressed in a text. It is commonly used to analyze customer feedback, survey responses, and product reviews. Social media monitoring, reputation management, and customer experience are just a few areas that can benefit from sentiment analysis.

For the EWS project sentiment classification shall be made in three different languages: English, Spanish and Tagalog.

Sentiment analysis in English has been implemented using `analyze_sentiment`. This model has been selected because it is already available as a spark NLP migrated model and its enhanced performance on large datasets make it suitable for EWS project.

Sentiment Analysis in Spanish has been implemented using `classifierdl_bert_sentiment` pretrained model. This model has been selected because it is already available as a spark NLP migrated model and its enhanced performance on large datasets make it suitable for EWS project.

Due to the lack of sentiment analysis pretrained model in such a low-resource language as Tagalog (no dataset or model has been found), translating Tagalog to English and using the model defined at 1.3.1.2 section will be our preferred solution.





1.4.1 Sentiment analysis in English 1.4.1.1

Model description

Model Name:	analyze_sentiment
Compatibility:	Spark NLP 3.0.0+
License:	Open Source
Edition:	Official
Input Labels:	[sentence_embeddings]
Output Labels:	[POSITIVE ,NEUTRAL,NEGATIVE]
Language:	en

1.4.1.2. Implementation

```
document = DocumentAssembler() \
.setInputCol("text") \
.setOutputCol("document")
token = Tokenizer() \
.setInputCols(["document"]) \ .setOutputCol("token")

normalizer = Normalizer() \
.setInputCols(["token"]) \ .setOutputCol("normal")

vivekn = ViveknSentimentModel.pretrained() \
.setInputCols(["document", "normal"]) \ .setOutputCol("result_sentiment")

finisher = Finisher() \
.setInputCols(["result_sentiment"]) \ .setOutputCols("final_sentiment") pipeline =
Pipeline().setStages([document, token, normalizer, vivekn, finisher])
```





1.4.1.3 Examples

Sentence	Sentiment
I really love her smile	POSITIVE
She always makes me laugh	POSITIVE
I usually have breakfast at 8	NEUTRAL
I am fed up with his behaviour	NEGATIVE

1.4.2 Sentiment analysis in Spanish

1.4.2.1 Model description

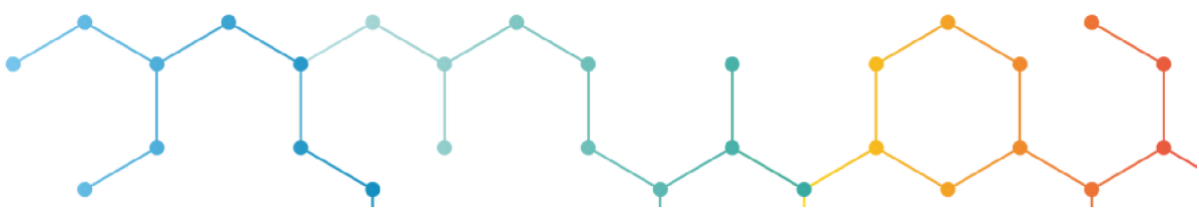
Model Name:	classifierdl_bert_sentiment
Compatibility:	Spark NLP 3.3.0+
License:	Open Source
Edition:	Official

Input Labels:	[sentence_embeddings]
Output Labels:	[POSITIVE,NEUTRAL,NEGATIVE]
Language:	es

1.4.2.2 Implementation

```
document = DocumentAssembler()\
.setInputCol("text")\
.setOutputCol("document")
```

```
embeddings = BertSentenceEmbeddings\
.pretrained('labse', 'xx') \
.setInputCols(["document"])\
```





```
.setOutputCol("sentence_embeddings")
```

```
sentimentClassifier = ClassifierDLModel.pretrained("classifierdl_bert_sentiment", "es") \
.setInputCols(["document", "sentence_embeddings"]) \
.setOutputCol("class") fr_sentiment_pipeline = Pipeline(stages=[document, embeddings,
sentimentClassifier])
```

1.4.2.3 Examples

Sentence	Sentiment
Estoy seguro de que esta vez pasará la entrevista	POSITIVE
Soy una persona que intenta desayunar todas las mañanas sin falta	NEUTRAL
No estoy seguro de si mi salario mensual es suficiente para vivir	NEGATIVE

1.5 Topic modelling

Topic modelling is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. Topic modelling is a frequently used text-mining tool for the discovery of hidden semantic structures in a text body.

In the age of information, the amount of written material we encounter each day is simply beyond our processing capacity. Topic models can help to organize and offer insights for us to understand large collections of unstructured text bodies.

In the EWS project, once all the tweets have been classified according to their main topic thanks to the tags included during the search, i.e. education, violence, reproductive rights, work and politics, each of the topics are subdivided as well into different categories according to the following chart:

Topic	Subtopics
Education	"Policy", "Career", "Money", "Diversity", "Other"
Violence	"Sexual Violence", "Racism", "Laws", "Crime", "Other"
Reproductive Rights	"Abortion", "HIV", "Crime", "LGBT", "Other"
Work	"Wealth", "Gender inequality", "Stereotypes", "Success", "Other"
Politics	"Emigration", "Public finance", "Leadership", "Violence", "Other"





The list of subtopics has been identified in the first stages of the project, where a set of 5000 tweets per topic and country were analysed and a list of the most common terms used was created for further analysis.

Once the subtopic classification was implemented, since there is no available dataset that could classify all of the topics at once, first a Latent Dirichlet Allocation (LDA) model was used to predict the different subyacent topics. This approach did not provide the expected results, so we explore other alternatives to get a meaningful subclassification. A state-ofthe-art algorithm used for unsupervised learning was selected to classify each tweet according the the different selected subcategories. This model is called zero-shot classifier.

1.5.1 Latent Dirichlet Allocation LDA

LDA represents documents as mixtures of topics that spit out words with certain probabilities. It assumes that documents are produced in the following fashion: when writing each document, you

- Decide on the number of words N the document will have (Poisson distribution).
- Choose a topic mixture for the document (according to a Dirichlet distribution over a fixed set of K topics).
- Generate each word w_i in the document by:
 - First picking a topic
 - Using the topic to generate the word itself (according to the topic's multinomial distribution).

Assuming this generative model for a collection of documents, LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated the collection.

1.5.1.1 Implementation

```
def TopicModelling(df):  
# Apply the function above and get tweets free of emoji's  call_emoji_free =  
lambda x: give_emoji_free_text(x)  
  
# Apply `call_emoji_free` which calls the function to remove all emoji's df['emoji_free_tweets'] =  
df['tweet'].apply(call_emoji_free)  
  
# Apply the function above and get tweets free of emoji's call_emoji_free =  
lambda x: deEmojify(x)  
  
# Apply `call_emoji_free` which calls the function to remove all emoji's df['emoji_free_tweets'] =  
df['emoji_free_tweets'].apply(call_emoji_free)  
  
# Apply the function above and get tweets free of emoji's call_emoji_free =  
lambda x: clean_tweet(x)
```





```
# Apply
`call_emoji_free` which calls the function to remove all emoji's df['emoji_free_tweets'] =
df['emoji_free_tweets'].apply(call_emoji_free)

#Create a new column with url free tweets
df['url_free_tweets'] = df['emoji_free_tweets'].apply(url_free_text)

# Load spacy
# Make sure to restart the runtime after running installations and libraries tab nlp =
spacy.load('en_core_web_lg')

# Tokenizer
tokenizer = Tokenizer(nlp.vocab)

stopwords.add("girl") #
Custom stopwords
custom_stopwords = ['hi','\n','\n\n','&',' ','!','-', 'got', 'it's', 'it's', 'i'm', 'i'm', 'im', 'want',
'like', '$', '@', 'girl', 'girls']

# Customize stop words by adding to the default list
STOP_WORDS = nlp.Defaults.stop_words.union(custom_stopwords)

# ALL_STOP_WORDS = spacy + gensim + wordcloud
_WORDS = nlp.Defaults.stop_words.union(custom_stopwords)

# ALL_STOP_WORDS = spacy + gensim + wordcloud
ALL_STOP_WORDS = STOP_WORDS.union(SW).union(stopwords)

tokens = []

for doc in tokenizer.pipe(df['url_free_tweets'], batch_size=10000):
    doc_tokens = [] for token in doc:
        if token.text.lower() not in STOP_WORDS:
            doc_tokens.append(token.text.lower()) tokens.append(doc_tokens)

# Makes tokens column df['tokens']
= tokens

# Make tokens a string again df['tokens_back_to_text'] = ['
'.join(map(str, l)) for l in df['tokens']]

df['lemmas'] = df['tokens_back_to_text'].apply(get_lemmas)
# Make lemmas a string again df['lemmas_back_to_text'] = ['
'.join(map(str, l)) for l in df['lemmas']]
```





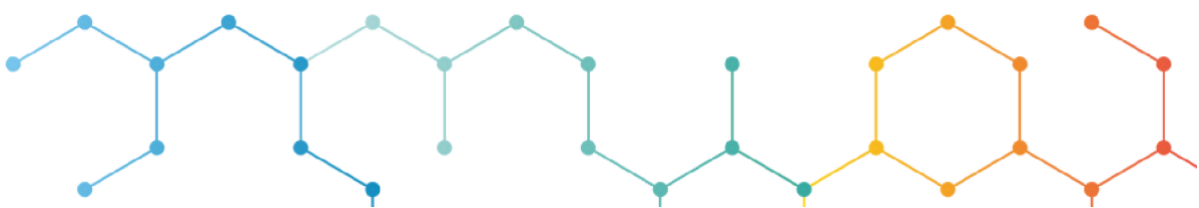
```
# Apply
tokenizer df['lemma_tokens'] =
df['lemmas_back_to_text'].apply(tokenize)

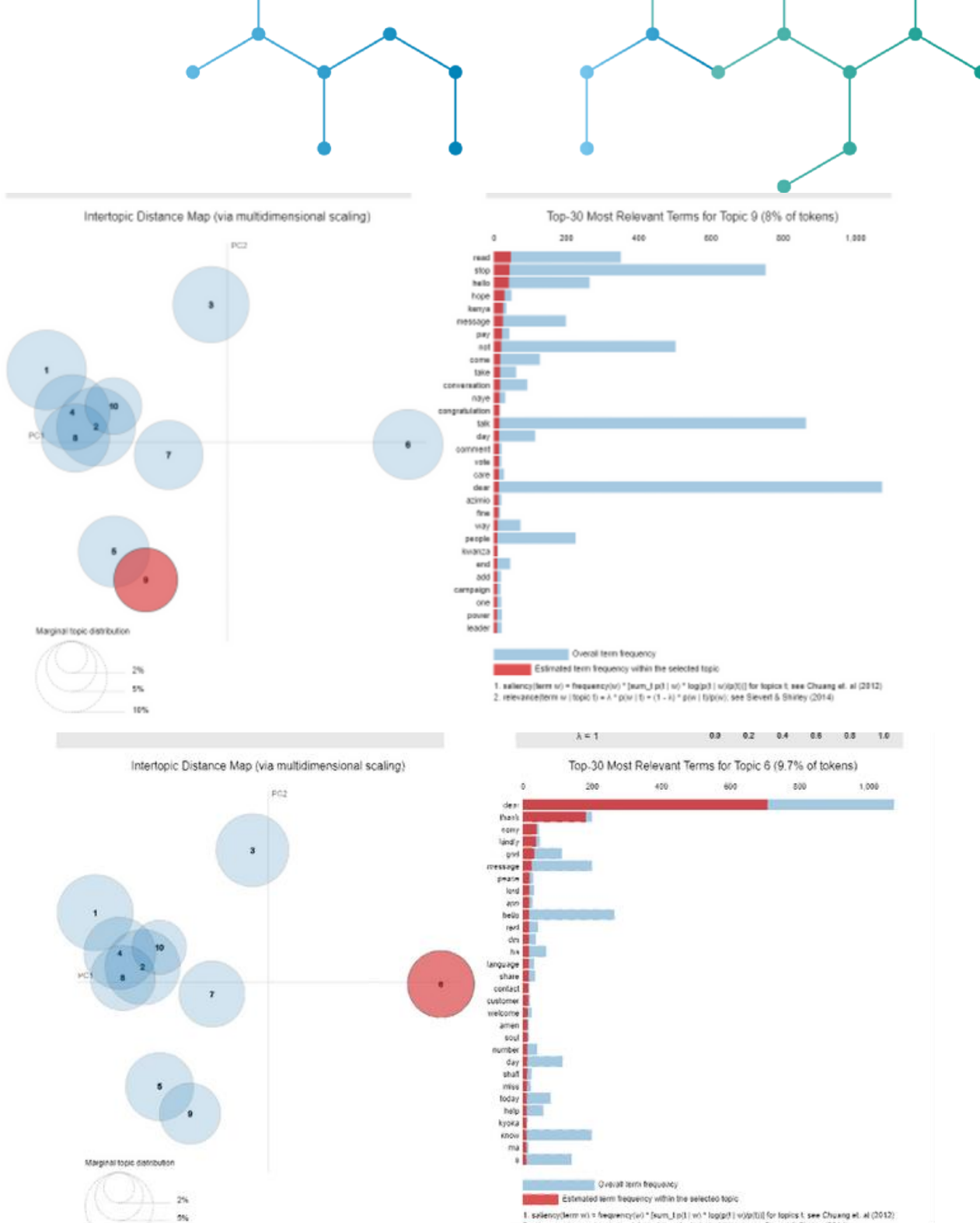
# Create a id2word dictionary id2word =
Dictionary(df['lemma_tokens'])
#print(len(id2word)) #
Filtering Extremes
id2word.filter_extremes(no_below=2, no_above=.99)
#print(len(id2word)) #
Creating a corpus object
corpus = [id2word.doc2bow(d) for d in df['lemma_tokens']]
# Instantiating a Base LDA model base_model =
LdaMulticore(corpus=corpus,
              id2word=id2word,
              num_topics=10,
              random_state=42,
              chunksize=2000,
              passes=25,          decay=0.5,
              iterations=70)
# Filtering for words
words = [re.findall(r"([^\s]*)",t[1]) for t in base_model.print_topics()]
# Create Topics
topics = [' '.join(t[0:10]) for t in words]
# Getting the topics
#for id, t in enumerate(topics):
# print(f"----- Topic {id} -----")
# print(t, end="\n\n")

#Creating Topic Distance Visualization pyLDAvis.enable_notebook()
pyLDAvis.gensim_models.prepare(base_model, corpus, id2word) p =
pyLDAvis.gensim_models.prepare(base_model, corpus, id2word) return p
```

1.5.1.2 Example

In this example, made to classify hate speech in Uganda, there are five distinct topics that can be appreciated, which correspond to the most distant bubbles (3,6,7, cluster 5 -9 and cluster 1-2-4-8). For each topic, we get a list of the 30 most relevant terms but as we can see from figures in the next figures, the list of relevant words on their own does not provide a very precise topic.





1.5.2 Zero-shot classification

Zero-shot learning (ZSL) most often referred to a specific type of task: learn a classifier on one set of labels and then evaluate on a different set of labels that the classifier has never seen before. Recently, especially in NLP, it's been used much more broadly to mean get a model to do something that it wasn't explicitly trained to do.

In zero-shot classification, you can define your own labels and then run classifier to assign a probability to each label. There is an option to do multi-class classification too, in this case, the scores will be independent, each will fall between 0 and 1.





Unsupervised text classification with zero-shot model allows us to solve text sentiment detection tasks when you don't have training data to train the model. Instead, you rely on a large trained model from transformers. For specialized use cases, when text is based on specific words or terms — is better to go with a supervised classification model, based on the training set. But for general topics, zero-shot model Works really well.

For the EWS project, the zero-shot classifier selected was released in hugging face under the name 'facebook/bart-large-mnli'[1]. The underlying model is trained on the task of Natural Language Inference (NLI), which takes in two sequences and determines whether they contradict each other, entail each other, or neither.

This can be adapted to the task of zero-shot classification by treating the sequence which we want to classify as one NLI sequence (called the premise) and

[1] <https://huggingface.co/facebook/bart-large-mnli>

turning a candidate label into the other (the hypothesis). If the model predicts that the constructed premise entails the hypothesis, then we can take that as a prediction that the label applies to the text.

By default, the pipeline turns labels into hypotheses with the template This example is {class_name}.. This works well in many settings, but you can also customize this for your specific setting.

In this case, all the topic modelling has been implemented in English, all the Spanish and Tagalog tweets were translated into English using the models defined at 4.1.1.

1.5.2.1 Implementation

```
from transformers import pipeline
import pandas as pd
```

```
df_hate = df_pandas[df_pandas["HateSpeech"]=="1"] df_hate["Subtopic"] =
```

```
subtopics = [
    ["Policy", "Career", "Money", "Diversity", "Other"],
    ["Carrer", "Business", "Scholarships", "Coding", "Other"],
    ["Sexual Violence", "Racism", "Laws", "Crime", "Other"],
    ["Abortion", "HIV", "Crime", "LGBT", "Other"],
    ["Wealth", "Gender inequality", "Stereotypes", "Success", "Other"],
    ["Emigration", "Public finance", "Leadership", "Violence", "Other"]
]
```

```
classifier = pipeline("zero-shot-classification")
```

```
for index, row in df_hate.iterrows():
    sequence = row["text"]
```





```

output =
classifier(sequence, subtopics[row["Topic"]-1])  if(output["scores"][0]>=0.5):
    df_hate.at[index,"Subtopic"] = output["labels"][0]
else:    df_hate.at[index,"Subtopic"] = "Other"

```

1.5.2.2 Example

says she was abducted held incommunicado tortured raped she goes ahead name perpetrators	Sexual violence
men are equal sight god his kingdom they however carry different responsibilities assigned	Stereotypes

Sentence	Topic
hey did you know that financial inclusion significantly contributes economic empowerment gender	Gender inequality





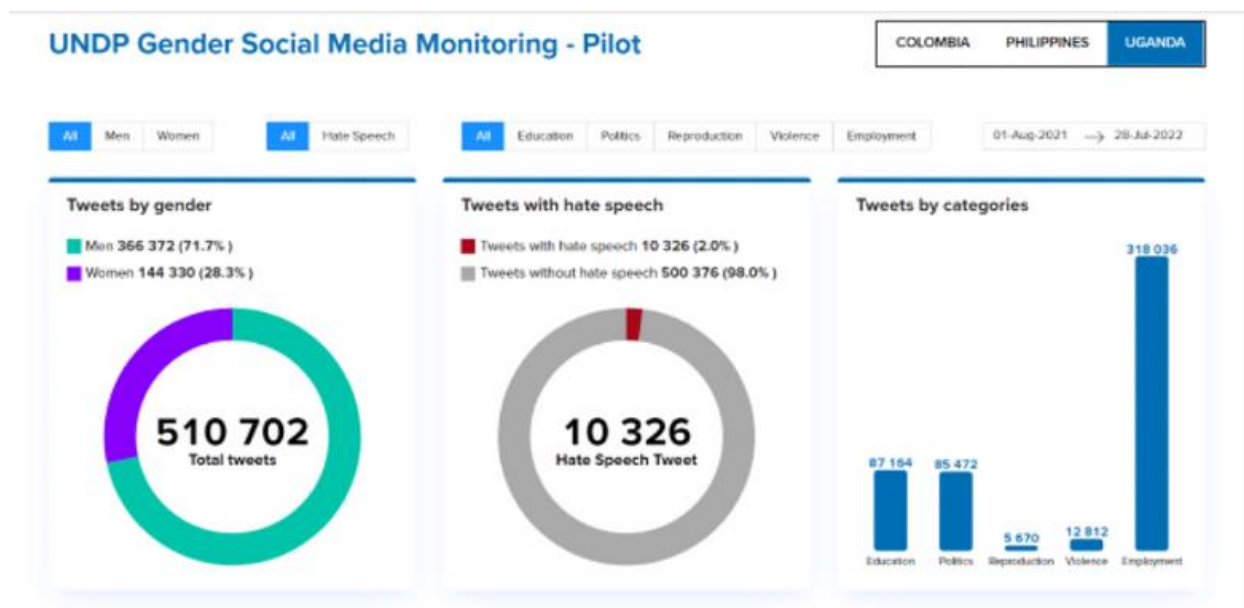


Conclusion

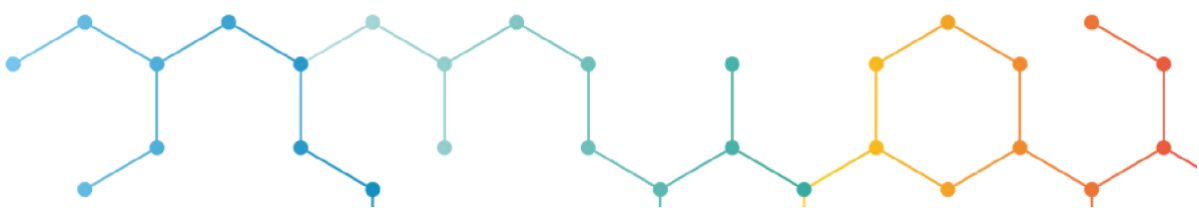
1.1 Conclusion

Due to the large scope of the EWS, this has undergone many changes throughout its life cycle. Numerous challenges have been faced, starting with the methodology, which has had to be refined up to four times, going through the inference process using big data models in distributed environments that have had to be reviewed on numerous occasions due to the great consumption of time and resources they involved.

From the technical point of view, the most difficult part of the project has occurred during the data batch analysis phase. Prediction processes of more than two million entries have had to be managed, in which the project's hardware resources have had to be scaled, the software has had to be migrated to new cluster management technologies and other distributed environments, and even thus the processing time has been quite large.



Despite the difficulties, it has been possible to extract some relevant data from our analysis, such as the correlation of the increase in hate speech under certain categories.





1.2 Carbon footprint

Due to the need for a historical analysis of the data that extends over a year, a large amount of information had to be handled in this project. Training new models based on big data and making predictions consumes a lot of resources, so the carbon footprint associated with the development of this tool is not negligible.

Below is an approximate calculation of the resources that have been used to carry out the first phase of this project:

Processing unit	Total Time used (hours)	Carbon emitted (kg CO ₂ eq) ¹⁶
Azure Advance_analysis cluster	1440	159.84
Azure Analysis_GPU cluster	2160	199.8
Azure Analysis_CPU cluster	480	26.64
UN global pulse	396	21.98
		408.26

Below is an approximate calculation of the resources that have been used to carry out the first phase of this project:


Processing unit	Total Time used (hours)	Carbon emitted (kg CO ₂ eq) ¹⁷
Azure Analysis_GPU cluster	250	23.125
UN global pulse	216	11.98
		35.105

Additionally, for phase I, the carbon footprint for the Azure Databricks continuous pipeline that runs daily to update the data corresponds to:

Processing unit	Total Time used (hours)	Carbon emitted (kg CO ₂ eq) ¹⁷
Azure Analysis_GPU cluster	0.6	0.0555

For phase II, the carbon footprint for the Azure Databricks continuous pipeline that runs daily to update the data corresponds to:

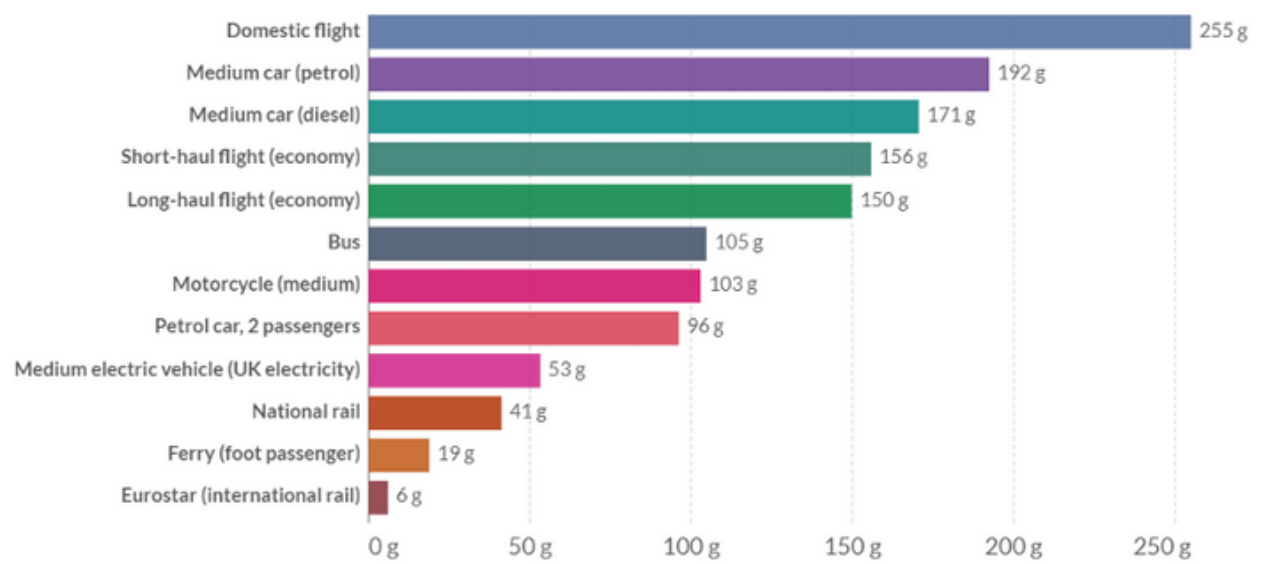




Processing unit	Total Time used (hours)	Carbon emitted (kg CO ₂ eq) ¹⁹
Azure Analysis_GPU cluster	0.25	0.023

The following table shows the carbon footprint of travel per kilometer, providing some insights to put the calculated values into context:

[1] Calculations made on <https://mlco2.github.io/impact/>
[2] Calculations made on <https://mlco2.github.io/impact/>





Glossary

API	Application Programming Interface
EWS	Early Warning System
LDA	Latent Dirichlet Allocation
NLP	Natural Language Processing
UN	United Nations

References

- [1] Riofrio, C. E., Chóez, A. C., & Gamboa, J. Z. (2021). Métodos de extracción de comentarios de la red social Twitter para uso en Procesamiento de Lenguaje Natural. Polo del Conocimiento: Revista científico-profesional, 6(11), 104-123.
- [2] Fantinuoli, C. (2016). Revisiting corpus creation and analysis tools for translation tasks. Cadernos de Tradução, 36(1), 62. <https://doi.org/10.5007/2175-7968.2016v36nesp1p62>
- [3] Han, B., Cook, P., & Baldwin, T. (2014). Text-based twitter user geolocation prediction. Journal of Artificial Intelligence Research, 49, 451–500. <https://doi.org/10.1613/jair.4200>
- [4] Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021, January). Lakehouse:





a new generation of open platforms that unify data warehousing and advanced analytics. In Proceedings of CIDR.

[5] Burger, J. D., Henderson, J., Kim, G., & Zarrella, G. (2011). Discriminating gender on Twitter. MITRE CORP BEDFORD MA BEDFORD United States.

[6] Ludu, P. S. (2014). Inferring gender of a Twitter user using celebrities it follows. arXiv preprint arXiv:1405.6667.

[7] Verhoeven, B., Daelemans, W., & Plank, B. (2016). Twisty: a multilingual twitter stylometry corpus for gender and personality profiling. In Proceedings of the

10th Annual Conference on Language Resources and Evaluation (LREC 2016)/Calzolari, Nicoletta [edit.]; et al. (pp. 1-6).

[8] Qudar, M. M. A., & Mago, V. (2020). Tweetbert: a pretrained language representation model for twitter text analysis. arXiv preprint arXiv:2010.11091.

[9] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[10] Aluru, S. S., Mathew, B., Saha, P., & Mukherjee, A. (2020). Deep learning models for multilingual hate speech detection. arXiv preprint arXiv:2004.06465.

[11] Hateful posts on Facebook and Instagram soar. Fortune. Retrieved 2020-11-21.

[12] Vidgen, B., Botelho, A., Broniatowski, D., Guest, E., Hall, M., Margetts, H., ... & Hale, S. (2020). Detecting East Asian prejudice on social media. arXiv preprint arXiv:2005.03909.





Appendix A: Batch data creation code

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:

import os
import requests
import datetime
import json

# In[21]:

download_year = '2022'

# In[22]:

timeframe_start = '{}08010000'.format(download_year)
timeframe_end = '{}08310000'.format(download_year)

# In[23]:

query_edu_Uganda = '(woman OR women OR girl OR girls OR female OR
females OR widow OR widows OR mother OR mothers OR wife OR wives OR
girlfriend OR girlfriends) (education OR school OR schools OR
educate OR edu OR college OR uni OR university OR teacher OR teachers
OR learning OR course OR teaching) (point_radius:[32.582520 0.347596
25mi] OR place:"Kampala" OR (profile_country:UG ))'
query_stem_Uganda = '(woman OR women OR girl OR girls OR female OR
females OR widow OR widows OR mother OR mothers OR wife OR wives OR
girlfriend OR girlfriends) (STEM OR hacker OR science OR code OR
coding OR technology OR engineering OR mathematics OR tech)
(point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR
(profile_country:UG ))'
query_violence_Uganda = '(woman OR women OR girl OR girls OR female
OR females OR widow OR widows OR mother OR mothers OR wife OR wives
OR girlfriend OR girlfriends) (rape OR sexual assault OR sexual
violence OR sexual abuse OR force sex OR child marriage OR children
marriage OR forced marriage OR sex trafficking OR child trafficking
```





```
OR children trafficking OR female genital mutilation OR female
genital cutting) (point_radius:[32.582520 0.347596 25mi] OR
place:"Kampala" OR (profile_country:UG ))'
query_reproductive_Uganda = '(woman OR women OR girl OR girls OR
female OR females OR widow OR widows OR mother OR mothers OR wife OR
wives OR girlfriend OR girlfriends) (abortion OR contraception OR
birth control OR pill OR IUD OR unwanted pregnancy)
(point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR
(profile_country:UG ))'
query_work_Uganda = '(woman OR women OR girl OR girls OR female OR
females OR widow OR widows OR mother OR mothers OR wife OR wives OR
girlfriend OR girlfriends) (work OR working OR career OR job OR
employment OR office OR employ OR employed OR employment OR ambition
OR success OR failure OR promotion OR promoted OR demotion OR demoted
OR salary OR raise OR pay OR child OR children OR kid OR kids OR
toddler OR baby OR babies OR infant OR infants OR family OR home OR
domestic ) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala"
OR (profile_country:UG ))'
query_political_Uganda = '(woman OR women OR girl OR girls OR female
OR females OR widow OR widows OR mother OR mothers OR wife OR wives
OR girlfriend OR girlfriends OR candidate) (lead OR leader OR leaders
OR leadership OR power OR powerful OR politics OR administration OR
administrations OR government OR governments OR state OR states OR
political party OR political parties OR gvt OR gov OR govt OR govmt
OR govmnt OR govnt OR politician OR politicians OR state funds OR
party funds OR public funds OR campaign promise OR campaign promises
OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresent OR
misrepresented OR misrepresentation) (point_radius:[32.582520
0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
query_edu_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR cunt
OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR
whore) (education OR school OR schools OR educate OR edu OR college
OR uni OR university OR teacher OR teachers OR learning OR course OR
teaching) (point_radius:[32.582520 0.347596 25mi] OR place:"Kampala"
OR (profile_country:UG ))'
query_stem_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR cunt
OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR
whore) (STEM OR hacker OR science OR code OR coding OR technology OR
engineering OR mathematics OR tech) (point_radius:[32.582520 0.347596
25mi] OR place:"Kampala" OR (profile_country:UG ))'
query_violence_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR
cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag
OR whore) (rape OR sexual assault OR sexual violence OR sexual abuse
OR force sex OR child marriage OR children marriage OR forced
marriage OR sex trafficking OR child trafficking OR children
trafficking OR female genital mutilation OR female genital cutting)
(point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR
(profile_country:UG ))'
query_reproductive_hate_Uganda = '(bimbo OR bitch OR cougar OR crone
OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR
wag OR whore) (abortion OR contraception OR birth control OR pill OR
```





```
IUD OR unwanted pregnancy) (point_radius:[32.582520 0.347596 25mi] OR
place:"Kampala" OR (profile_country:UG ))'
query_work_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR cunt
OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR
whore) (work OR working OR career OR job OR employment OR office OR
employ OR employed OR employment OR ambition OR success OR failure OR
promotion OR promoted OR demotion OR demoted OR salary OR raise OR
pay OR child OR children OR kid OR kids OR toddler OR baby OR babies
OR infant OR infants OR family OR home OR domestic )
(point_radius:[32.582520 0.347596 25mi] OR place:"Kampala" OR
(profile_country:UG ))'
query_political_hate_Uganda = '(bimbo OR bitch OR cougar OR crone OR
cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag
OR whore) (lead OR leader OR leaders OR leadership OR power OR
powerful OR politics OR administration OR administrations OR
government OR governments OR state OR states OR political party OR
political parties OR gvt OR gov OR govt OR govmt OR govmnt OR govtnt
OR politician OR politicians OR state funds OR party funds OR public
funds OR campaign promise OR campaign promises OR corrupt OR
corruption OR vote OR votes OR fraud OR misrepresent OR
misrepresented OR misrepresentation) (point_radius:[32.582520
0.347596 25mi] OR place:"Kampala" OR (profile_country:UG ))'
```

```
# In[24]:
```

```
query_edu_Colombia = '(mujer OR mujeres OR niÃ±a OR niÃ±as OR mujer
OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas
OR novia OR novias) (educaciÃ³n OR escuela OR escuelas OR colegio OR
colegios OR educar OR educaciÃ³n OR universidad OR universidades OR
uni OR maestro OR maestros OR profesor OR profesores OR aprendizaje
OR curso OR enseÃ±anza) (point_radius:[-74.063644 4.624335 25mi] OR
place:"Bogota" OR (profile_country:CO ))'
query_stem_Colombia = '(mujer OR mujeres OR niÃ±a OR niÃ±as OR mujer
OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas
OR novia OR novias) (STEM OR ciencia OR cÃ³digo OR codificar OR
coding OR hacker OR coder OR tecnologÃa OR tech OR ingenierÃa OR
matemÃticas OR tech) (point_radius:[-74.063644 4.624335 25mi] OR
place:"Bogota" OR (profile_country:CO ))'
query_violence_Colombia = '(mujer OR mujeres OR niÃ±a OR niÃ±as OR
mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR
esposas OR novia OR novias) (violaciÃ³n OR feminicidio OR violencia
vicaria OR violencia machista OR violencia de gÃnero OR asesinato OR
agresiÃ³n sexual OR manada OR abuso sexual OR abusos sexuales OR sexo
sin consentimiento OR matrimonio infantil OR matrimonios infantiles
OR matrimonio forzado OR trÃfico de mujeres OR trÃfico de niÃ±os OR
secuestro) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota"
OR (profile_country:CO ))'
query_reproductive_Colombia = '(mujer OR mujeres OR niÃ±a OR niÃ±as
OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa
OR esposas OR novia OR novias) (aborto OR anticonceptivos OR control
```





```
de natalidad OR pãldora OR DIU OR embarazo no deseado)
(point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR
(profile_country:CO ))'
query_work_Colombia = '(mujer OR mujeres OR niãta OR niãtas OR mujer
OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas
OR novia OR novias) (trabajo OR trabajando OR empleo OR carrera OR
puesto OR oficina OR emplear OR empleado OR empleador OR ambiciã³n OR
ãxito OR fracaso OR promociã³n OR ascenso OR ascendido OR descenso
OR relegar OR relegada OR salario OR subida salarial OR pagar OR
niãto OR niãtos OR bebã© OR bebã©s OR menor OR menores OR familia OR
casa OR hogar OR domãstico ) (point_radius:[-74.063644 4.624335
25mi] OR place:"Bogota" OR (profile_country:CO ))'
query_political_Colombia = '(mujer OR mujeres OR niãta OR niãtas OR
mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR
esposas OR novia OR novias OR gobernadora OR candidata) (liderar OR
lãder OR lãderes OR liderazgo OR poder OR poderoso OR polãtica OR
administraciã³n OR administraciones OR gobierno OR gobiernos OR
estado OR estados OR partido polãtico OR partidos polãticos OR
gobierno O gobierno O gobierno O gobierno O gobierno OR fondos
estatales OR fondos del partido OR fondos pãblicos OR promesa
electoral OR promesa de campãta OR corrupciã³n O corrupto OR voto OR
votos OR fraude OR tergiversaciã³n OR representaciã³n)
(point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR
(profile_country:CO ))'
query_edu_hate_Colombia = '(traidora OR corrupta OR asesina OR
ladrona OR cinica OR ridicula OR mentirosa OR pelotuda OR boluda OR
burra OR mafiosa OR estúpida OR gorda OR bruta OR patetica OR enferma
OR payasa OR inmundada OR fracasada OR siniestra OR loca OR tarada OR
puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR
rastrera OR estafadora OR resentida OR mugrienta OR tonta OR guarra
OR zorra OR calientapollas)(educaciã³n OR escuela OR escuelas OR
colegio OR colegios OR educar OR educaciã³n OR universidad OR
universidades OR uni OR maestro OR maestros OR profesor OR profesores
OR aprendizaje OR curso OR enseãtanza) (point_radius:[-74.063644
4.624335 25mi] OR place:"Bogota" OR (profile_country:CO ))'
query_stem_hate_Colombia = '(traidora OR corrupta OR asesina OR
ladrona OR cinica OR ridicula OR mentirosa OR pelotuda OR boluda OR
burra OR mafiosa OR estúpida OR gorda OR bruta OR patetica OR enferma
OR payasa OR inmundada OR fracasada OR siniestra OR loca OR tarada OR
puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR
rastrera OR estafadora OR resentida OR mugrienta OR tonta OR guarra
OR zorra OR calientapollas)(STEM OR ciencia OR cã³digo OR codificar
OR coding OR hacker OR coder OR tecnologãa OR tech OR ingenierãa OR
matemã;ticas OR tech) (point_radius:[-74.063644 4.624335 25mi] OR
place:"Bogota" OR (profile_country:CO ))'
query_violence_hate_Colombia = '(traidora OR corrupta OR asesina OR
ladrona OR cinica OR ridicula OR mentirosa OR pelotuda OR boluda OR
burra OR mafiosa OR estúpida OR gorda OR bruta OR patetica OR enferma
OR payasa OR inmundada OR fracasada OR siniestra OR loca OR tarada OR
puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR
rastrera OR estafadora OR resentida OR mugrienta OR tonta OR guarra
OR zorra OR calientapollas)(violaciã³n OR feminicidio OR violencia
```





```
vicaria OR violencia machista OR violencia de género OR asesinato OR
agresión sexual OR manada OR abuso sexual OR abusos sexuales OR sexo
sin consentimiento OR matrimonio infantil OR matrimonios infantiles
OR matrimonio forzado OR tráfico de mujeres OR tráfico de niños OR
secuestro) (point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota"
OR (profile_country:CO ))'
query_violence_hate_Colombia_ext = '(abombada OR absolutista OR
agrogarca OR alcahueta OR altanera OR alzada OR amoral OR analfabeta
OR anda a cagar OR anda a la mierda OR anda a terapia OR argolluda OR
arpiá OR arrastrada OR asesina OR asquerosa OR atorranta OR atrevida
OR banate OR bastarda OR basura OR berreta OR bicha OR bobeta OR
boluda OR borracha OR bosta OR braguetera OR buena para nada OR burda
OR burra OR cabeza hueca OR cacatía OR cachavache OR cagadora OR
cagona OR calandraca OR caprichosa OR cara de OR casta inmunda OR
catadora de waska OR cerda OR chapita OR chupasangre OR cornuda OR
conventillera OR culo roto OR das asco OR das pena OR das repulsión
OR degenerado OR gorda OR grosera OR gilipolla OR hacete coger OR
hacete humo) (point_radius:[-74.063644 4.624335 25mi] OR
place:"Bogota" OR (profile_country:CO ))'
query_reproductive_hate_Colombia = '(traidora OR corrupta OR asesina
OR ladrona OR cinica OR ridicula OR mentirosa OR pelotuda OR boluda
OR burra OR mafiosa OR estúpida OR gorda OR bruta OR patetica OR
enferma OR payasa OR inmunda OR fracasada OR siniestra OR loca OR
tarada OR puta OR descerebrada OR tibia OR asquerosa OR feminazi OR
boba OR rastrera OR estafadora OR resentida OR mugrienta OR tonta OR
guarra OR zorra OR calientapollas) (aborto OR anticonceptivos OR
control de natalidad OR píldora OR DIU OR embarazo no deseado)
(point_radius:[-74.063644 4.624335 25mi] OR place:"Bogota" OR
(profile_country:CO ))'
query_work_hate_Colombia = '(traidora OR corrupta OR asesina OR
ladrona OR cinica OR ridicula OR mentirosa OR pelotuda OR boluda OR
burra OR mafiosa OR estúpida OR gorda OR bruta OR patetica OR enferma
OR payasa OR inmunda OR fracasada OR siniestra OR loca OR tarada OR
puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR
rastrera OR estafadora OR resentida OR mugrienta OR tonta OR guarra
OR zorra OR calientapollas) (trabajo OR trabajando OR empleo OR
carrera OR puesto OR oficina OR emplear OR empleado OR empleador OR
ambición OR éxito OR fracaso OR promoción OR ascenso OR ascendido
OR descenso OR relegar OR relegada OR salario OR subida salarial OR
pagar OR niño OR niños OR bebé OR bebés OR menor OR menores OR
familia OR casa OR hogar OR doméstico ) (point_radius:[-74.063644
4.624335 25mi] OR place:"Bogota" OR (profile_country:CO ))'
query_work_hate_Colombia_ext = '(adocotrínada OR alienada OR anda a
laburar OR autoritaria OR bajate del pony OR chamuyera OR chanta OR
codiciosa OR curradora OR currera OR demagoga OR difamadora OR
desubicada OR esclavista OR irrespetuosa OR ladrona OR mafiosa OR
manipuladora OR mentirosa OR mercenaria OR mosquita muerta OR obtusa
OR perroncha OR radicheta OR vendepatria OR vieja) (point_radius:[-
74.063644 4.624335 25mi] OR place:"Bogota" OR (profile_country:CO ))'
query_politics_hate_Colombia = '(traidora OR corrupta OR asesina OR
ladrona OR cinica OR ridicula OR mentirosa OR pelotuda OR boluda OR
burra OR mafiosa OR estúpida OR gorda OR bruta OR patetica OR enferma
```





```
OR payasa OR inmundada OR fracasada OR siniestra OR loca OR tarada OR  
puta OR descerebrada OR tibia OR asquerosa OR feminazi OR boba OR  
rastrera OR estafadora OR resentida OR mugrienta OR tonta OR guarra  
OR zorra OR calientapollas)(liderar OR l der OR l deres OR liderazgo  
OR poder OR poderoso OR pol tica OR administraci n OR  
administraciones OR gobierno OR gobiernos OR estado OR estados OR  
partido pol tico OR partidos pol ticos OR gobierno O gobierno O  
gobierno O gobierno O gobierno OR fondos estatales OR fondos del  
partido OR fondos p blicos OR promesa electoral OR promesa de  
campa a OR corrupci n O corrupto OR voto OR votos OR fraude OR  
tergiversaci n OR representaci n) (point_radius:[-74.063644  
4.624335 25mi] OR place:"Bogota" OR (profile_country:CO ))'  
query_politics_hate_Colombia_ext = '(antirepublicana OR maleducada OR  
maricona OR milf OR militonta OR mononeuronal OR ordinaria OR  
perversa OR psicopata OR repulsiva OR ramera OR repulsiva OR  
resentida OR retardada OR terrorista OR traicionera OR vibora OR  
ventajera OR violenta OR vomito con patas) (point_radius:[-74.063644  
4.624335 25mi] OR place:"Bogota" OR (profile_country:CO ))'
```

```
# In[25]:
```

```
query_edu_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR  
babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina  
OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR  
misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR  
ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra  
OR negrita OR puta OR amputa OR shuta)(edukasyon OR paaralan OR mga  
paaralan OR edu OR kolehiyo OR unibersidad OR guro OR pag-aaral OR  
kurso OR pagtuturo)'  
query_stem_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR  
babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina  
OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR  
misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR  
ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra  
OR negrita OR puta OR amputa OR shuta) (STEM OR hacker OR science OR  
code OR coding OR teknolohiya OR engineering OR matematika OR tech)'  
query_violence_Philippines = '(ale OR ate OR ditse OR sanse OR sitse  
OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR  
filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR  
kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR  
neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida  
OR negra OR negrita OR puta OR amputa OR shuta) (panggagahasa OR  
sekswal na pag-atake OR sekswal na karahasan OR sekswal na pang-  
aabuso OR puwersahin ang sex OR child marriage OR kasal ng mga bata  
OR forced marriage OR sex trafficking OR child trafficking OR child  
trafficking) '  
query_reproductive_Philippines = '(ale OR ate OR ditse OR sanse OR  
sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR  
filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR  
kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR
```





```
neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida  
OR negra OR negrita OR puta OR amputa OR shuta) (pagpapalaglag OR  
pagpipigil sa pagbubuntis OR birth control OR tableta OR IUD OR hindi  
gustong pagbubuntis) '
```

```
query_work_Philippines = '(ale OR ate OR ditse OR sanse OR sitse OR  
babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR filipina  
OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR kumare OR  
misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR neneng OR  
ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida OR negra  
OR negrita OR puta OR amputa OR shuta) (trabaho OR nagtatrabaho OR  
karera OR trabaho OR trabaho OR opisina OR trabaho OR trabaho OR  
trabaho OR ambisyon OR tagumpay OR kabiguan OR promosyon OR na-  
promote OR pagbabawas OR pagbabawas OR pagbabawas OR pagbabawas OR  
pagbabawas OR pagbabawas OR pagbabawas OR pagbabawas OR pagpapababa  
OR suweldo OR pagtaas OR pagbabayad OR anak OR bata OR bata OR bata  
OR sanggol OR sanggol OR sanggol OR sanggol OR sanggol OR pamilya OR  
tahanan OR domestic )'
```

```
query_political_Philippines = '(ale OR ate OR ditse OR sanse OR sitse  
OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR  
filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR  
kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR  
neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida  
OR negra OR negrita OR puta OR amputa OR shuta) (pinuno OR pinuno OR  
pinuno OR pamumuno OR kapangyarihan OR makapangyarihan OR pulitika OR  
administrasyon OR mga administrasyon OR pamahalaan OR pamahalaan OR  
estado OR estado OR partidong pampulitika OR partidong pampulitika OR  
gvt OR gobyerno, OR gobyerno, OR pamahalaan, OR politiko OR mga  
pulitiko OR mga pondo ng estado OR mga pondo ng partido OR mga pondo  
ng publiko OR mga pangako sa kampanya OR mga pangako ng kampanya OR  
mga tiwali OR katiwalian OR bumoto OR mga boto OR pandaraya OR maling  
representasyon OR maling representasyon OR maling representasyon)'
```

```
# In[26]:
```

```
query_edu_hate_Philippines = '(ale OR ate OR ditse OR sanse OR sitse  
OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR  
filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR  
kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR  
neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida  
OR negra OR negrita OR puta OR amputa OR shuta) (batsilyer OR diploma  
OR dunong OR edukasyon OR estudyante OR grado OR kolehiyo OR paaralan  
OR pag-aaral OR pinag-aralan OR pagsusulit OR pagtuturo OR turo OR  
unibersidad)'
```

```
query_stem_hate_Philippines = '(ale OR ate OR ditse OR sanse OR sitse  
OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR  
filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR  
kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR  
neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida  
OR negra OR negrita OR puta OR amputa OR shuta) (agham OR matematika  
OR inhinyeriya OR pag-linhinyero OR sipnayan OR teknolohiya)'
```





```

query_violence_hate_Philippines = '(ale OR ate OR ditse OR sanse OR
sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR
filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR
kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR
neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida
OR negra OR negrita OR puta OR amputa OR shuta)(Abuso OR Pang-aabuso
OR Atakeng Sekswal OR Dahas OR Karahasan OR Gahasa OR Panggagahasa OR
Halay OR Panghahalay OR Insesto OR Lapastangan OR Kalapastanganan OR
Lupit OR Kalupitan OR Molestiya OR Pangmomolestiya OR Pagpatay OR
Puwersa OR Pamumwersa OR Sekswal na karahasan)'

query_reproductive_hate_Philippines = '(ale OR ate OR ditse OR sanse
OR sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga
OR filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare
OR kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene
OR neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR
kerida OR negra OR negrita OR puta OR amputa OR shuta)(Aborsyon OR
Buntis OR Pagbubuntis OR Kontraseptibo OR Laglag OR Pagpapalaglag OR
Pagdadalantao OR Pagpapaagas OR Pagpigil sa panganganak OR Pamparegla
OR Panganganak)'

query_work_hate_Philippines = '(ale OR ate OR ditse OR sanse OR sitse
OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR
filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR
kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR
neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida
OR negra OR negrita OR puta OR amputa OR shuta)(Ambisyon OR Benepisyo
OR Empleyado OR Manggagawa OR Hanapbuhay OR Karanasan OR Karera OR
Kumpanya OR Opisina OR pag-angat OR pagbabawas OR Pagpapatalsik OR
pagsisisante OR promosyon OR sahod OR suweldo OR tanggalan OR
trabaho)'

query_political_hate_Philippines = '(ale OR ate OR ditse OR sanse OR
sitse OR babae OR gerlalu OR bebot OR binibini OR biyuda OR dalaga OR
filipina OR pinay OR gelpren OR ginang OR inday OR manang OR mare OR
kumare OR misis OR nanay OR inay OR ina OR mama OR ermat OR nene OR
neneng OR ineng OR tita OR tiyahin OR tiya OR bruha OR gaga OR kerida
OR negra OR negrita OR puta OR amputa OR shuta)(Administrasyon OR
balota OR boto OR eleksyon OR halalan OR kapangyarihan OR karapatan
OR katungkulan OR korapsyon OR lider OR opisyal OR pagsisilbi OR
pamahalaan OR gobyerno OR pamumuno OR pangangampanya OR partido OR
pinuno OR pulitika OR trapo)'

```

```
# In[27]:
```

```

query_edu_Philippines_EN = '(woman OR women OR girl OR girls OR
female OR females OR widow OR widows OR mother OR mothers OR wife OR
wives OR girlfriend OR girlfriends) (education OR school OR schools
OR educate OR edu OR college OR uni OR university OR teacher OR
teachers OR learning OR course OR teaching) (point_radius:[120.984222
14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'
query_stem_Philippines_EN = '(woman OR women OR girl OR girls OR
female OR females OR widow OR widows OR mother OR mothers OR wife OR

```





```
wifes OR girlfriend OR girlfriends) (STEM OR hacker OR science OR
code OR coding OR technology OR engineering OR mathematics OR tech)
(point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR
(profile_country:PH))'
query_violence_Philippines_EN = '(woman OR women OR girl OR girls OR
female OR females OR widow OR widows OR mother OR mothers OR wife OR
wifes OR girlfriend OR girlfriends) (rape OR sexual assault OR sexual
violence OR sexual abuse OR force sex OR child marriage OR children
marriage OR forced marriage OR sex trafficking OR child trafficking
OR children trafficking OR female genital mutilation OR female
genital cutting) (point_radius:[120.984222 14.599512 25mi] OR
place:"Manila" OR (profile_country:PH))'
query_reproductive_Philippines_EN = '(woman OR women OR girl OR girls
OR female OR females OR widow OR widows OR mother OR mothers OR wife
OR wifes OR girlfriend OR girlfriends) (abortion OR contraception OR
birth control OR pill OR IUD OR unwanted pregnancy)
(point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR
(profile_country:PH))'
query_work_Philippines_EN = '(woman OR women OR girl OR girls OR
female OR females OR widow OR widows OR mother OR mothers OR wife OR
wifes OR girlfriend OR girlfriends) (work OR working OR career OR job
OR employment OR office OR employ OR employed OR employment OR
ambition OR success OR failure OR promotion OR promoted OR demotion
OR demoted OR salary OR raise OR pay OR child OR children OR kid OR
kids OR toddler OR baby OR babies OR infant OR infants OR family OR
home OR domestic ) (point_radius:[120.984222 14.599512 25mi] OR
place:"Manila" OR (profile_country:PH))'
query_political_Philippines_EN = '(woman OR women OR girl OR girls OR
female OR females OR widow OR widows OR mother OR mothers OR wife OR
wifes OR girlfriend OR girlfriends OR candidate) (lead OR leader OR
leaders OR leadership OR power OR powerful OR politics OR
administration OR administrations OR government OR governments OR
state OR states OR political party OR political parties OR gvt OR gov
OR govt OR govmt OR govmnt OR govnt OR politician OR politicians OR
state funds OR party funds OR public funds OR campaign promise OR
campaign promises OR corrupt OR corruption OR vote OR votes OR fraud
OR misrepresent OR misrepresented OR misrepresentation)
(point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR
(profile_country:PH))'
query_edu_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR crone
OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR
wag OR whore) (education OR school OR schools OR educate OR edu OR
college OR uni OR university OR teacher OR teachers OR learning OR
course OR teaching) (point_radius:[120.984222 14.599512 25mi] OR
place:"Manila" OR (profile_country:PH))'
query_stem_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR crone
OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR
wag OR whore) (STEM OR hacker OR science OR code OR coding OR
technology OR engineering OR mathematics OR tech)
(point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR
(profile_country:PH))'
```





```
query_violence_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR  
crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR  
twat OR wag OR whore) (rape OR sexual assault OR sexual violence OR  
sexual abuse OR force sex OR child marriage OR children marriage OR  
forced marriage OR sex trafficking OR child trafficking OR children  
trafficking OR female genital mutilation OR female genital cutting)  
(point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR  
(profile_country:PH))'  
query_reproductive_hate_Philippines_EN = '(bimbo OR bitch OR cougar  
OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR  
twat OR wag OR whore) (abortion OR contraception OR birth control OR  
pill OR IUD OR unwanted pregnancy) (point_radius:[120.984222  
14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'  
query_work_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR crone  
OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR  
wag OR whore) (work OR working OR career OR job OR employment OR  
office OR employ OR employed OR employment OR ambition OR success OR  
failure OR promotion OR promoted OR demotion OR demoted OR salary OR  
raise OR pay OR child OR children OR kid OR kids OR toddler OR baby  
OR babies OR infant OR infants OR family OR home OR domestic )  
(point_radius:[120.984222 14.599512 25mi] OR place:"Manila" OR  
(profile_country:PH))'  
query_political_hate_Philippines_EN = '(bimbo OR bitch OR cougar OR  
crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR  
twat OR wag OR whore) (lead OR leader OR leaders OR leadership OR  
power OR powerful OR politics OR administration OR administrations OR  
government OR governments OR state OR states OR political party OR  
political parties OR gvt OR gov OR govt OR govmt OR govmt OR govt  
OR politician OR politicians OR state funds OR party funds OR public  
funds OR campaign promise OR campaign promises OR corrupt OR  
corruption OR vote OR votes OR fraud OR misrepresent OR  
misrepresented OR misrepresentation) (point_radius:[120.984222  
14.599512 25mi] OR place:"Manila" OR (profile_country:PH))'
```

```
# In[28]:
```

```
query_data = {  
    'dataFormat': 'activity_streams',  
    'fromDate': timeframe_start,  
    'toDate': timeframe_end,  
    'title': 'Gender speech historical tweets (Started at  
{})'.format(datetime.datetime.utcnow().isoformat()),  
    'rules': [{'value': query_edu_Uganda, 'tag': "Edu_Uganda"  
}, {'value': query_edu_hate_Uganda, 'tag': "Edu_Hate_Uganda" }, {'value':  
query_stem_Uganda, 'tag': "Stem_Uganda" }, {'value':  
query_stem_hate_Uganda, 'tag': "Stem_Hate_Uganda" }, {'value':  
query_violence_Uganda, 'tag': "Violence_Uganda" }, {'value':  
query_violence_hate_Uganda, 'tag': "Violence_Hate_Uganda" }, {'value':  
query_reproductive_Uganda, 'tag': "Reproduction_Uganda" }, {'value':  
query_reproductive_hate_Uganda, 'tag': "Reproduction Hate_Uganda"
```





```

},{'value': query_work_Uganda,'tag':"Work_Uganda" },{'value':
query_work_hate_Uganda,'tag':"Work_Hate_Uganda" },{'value':
query_political_Uganda,'tag':"Politics_Uganda" },{'value':
query_political_hate_Uganda,'tag':"Politics_Hate_Uganda" },{'value':
query_edu_Colombia,'tag':"Edu_Colombia" },{'value':
query_edu_hate_Colombia,'tag':"Edu_Hate_Colombia" },{'value':
query_stem_Colombia,'tag':"Stem_Colombia" },{'value':
query_stem_hate_Colombia,'tag':"Stem_Hate_Colombia" },{'value':
query_violence_Colombia,'tag':"Violence_Colombia" },{'value':
query_violence_hate_Colombia,'tag':"Violence_Hate_Colombia"
},{'value':
query_violence_hate_Colombia_ext,'tag':"Violence_Hate_Colombia_ext"
},{'value': query_reproductive_Colombia,'tag':"Reproduction_Colombia"
},{'value':
query_reproductive_hate_Colombia,'tag':"Reproduction_Hate_Colombia"
},{'value': query_work_Colombia,'tag':"Work_Colombia" },{'value':
query_political_Colombia,'tag':"Politics_Colombia" },{'value':
query_work_hate_Colombia,'tag':"Work_Hate_Colombia" },{'value':
query_work_hate_Colombia_ext,'tag':"Work_Hate_Colombia_ext"
},{'value':
query_politics_hate_Colombia,'tag':"Politics_Hate_Colombia"
},{'value':
query_politics_hate_Colombia_ext,'tag':"Politics_Hate_Colombia_ext"
},
        {'value': query_edu_Philippines,'tag':"Edu_Philippines"
},{'value': query_stem_Philippines,'tag':"Stem_Philippines"
},{'value': query_violence_Philippines,'tag':"Violence_Philippines"
},{'value':
query_reproductive_Philippines,'tag':"Reproduction_Philippines"
},{'value': query_work_Philippines,'tag':"Work_Philippines"
},{'value': query_political_Philippines,'tag':"Politics_Philippines"
},{'value': query_edu_Philippines_EN,'tag':"Edu_Phi_EN" },{'value':
query_stem_Philippines_EN,'tag':"Stem_Phi_EN" },{'value':
query_violence_Philippines_EN,'tag':"Violence_Phi_EN" },{'value':
query_reproductive_Philippines_EN,'tag':"Reproduction_Phi_EN"
},{'value': query_work_Philippines_EN,'tag':"Work_Phi_EN" },{'value':
query_political_Philippines_EN,'tag':"Political_Phi_EN" },{'value':
query_edu_hate_Philippines_EN,'tag':"Edu_Hate_Phi_EN" },{'value':
query_stem_hate_Philippines_EN,'tag':"Stem_Hate_Phi_EN" },{'value':
query_violence_hate_Philippines_EN,'tag':"Violence_Hate_Phi_EN"
},{'value':
query_reproductive_hate_Philippines_EN,'tag':"Reproduction_Hate
_Phi_EN" },{'value': query_work_hate_Philippines_EN,'tag':"Work_Hate
_Phi_EN" },{'value':
query_political_hate_Philippines_EN,'tag':"Political_Hate_Phi_EN"
},{'value': query_edu_hate_Philippines,'tag':"Edu_Hate_Phi"
},{'value': query_stem_hate_Philippines,'tag':"Edu_Hate_Phi"
},{'value': query_violence_hate_Philippines,'tag':"Violence_Hate
_Phi" },{'value':
query_reproductive_hate_Philippines,'tag':"Reproduction_Hate_Phi"
},{'value': query_work_hate_Philippines,'tag':"Work_Hate_Phi"

```





```
{,{'value': query_political_hate_Philippines,'tag':"Politics_Hate_Phi" }}
}

# # 1. Create a Historical Job

# In[29]:

response =
requests.post("https://api.flock.unglobalpulse.net/historical/powertrack/accounts/UN/publishers/twitter/jobs.json",
              json = query_data,
              auth=(os.getenv('FLOCK_USER'),
os.getenv('FLOCK_PASS'))))

# In[30]:

response_json = response.text

# In[31]:

response_dict = json.loads(response_json)

# In[32]:

print(json.dumps(response_dict, indent=2))

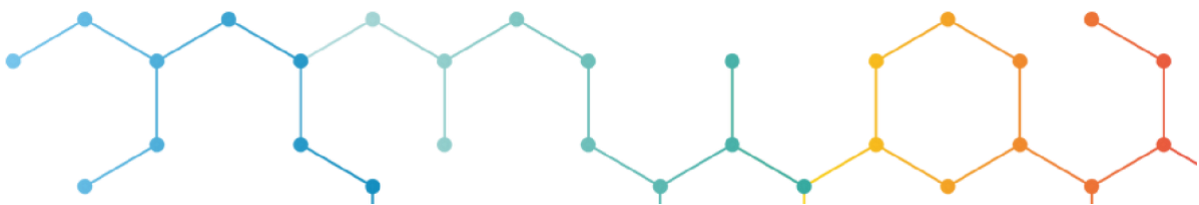
# # 2. Get Job Status While It's Estimating

# In[43]:

job_url =
"https://api.flock.unglobalpulse.net/historical/powertrack/accounts/UN/publishers/twitter/jobs/cw4zhby6s4.json"

# In[44]:

response_2 = requests.get(job_url,
                          auth=(os.getenv('FLOCK_USER'),
os.getenv('FLOCK_PASS'))))
```





```
# In[45]:

response_2_json = response_2.text

# In[46]:

response_2_dict = json.loads(response_2_json)

# In[47]:

print(json.dumps(response_2_dict, indent=2))

# # 3. Get Job Status After Quote is Available

# In[48]:

response_3 = requests.get(job_url,
                           auth=(os.getenv('FLOCK_USER'),
                                    os.getenv('FLOCK_PASS')))

# In[49]:

response_3_json = response_3.text

# In[50]:

response_3_dict = json.loads(response_3_json)

# In[51]:

print(json.dumps(response_3_dict, indent=2))

# # 4. Accept the Quoted Job

# In[52]:
```





```
accept_data = {
    'status': 'accept'
}

# In[53]:

response_4 = requests.put(job_url,
                           json = accept_data,
                           auth=(os.getenv('FLOCK_USER'),
os.getenv('FLOCK_PASS'))))

# In[54]:

response_4_dict = json.loads(response_4.text)

# In[55]:

print(json.dumps(response_4_dict, indent=2))

# # 5. Get the Accepted Running Job Status

# In[76]:

job_url =
"https://api.flock.unglobalpulse.net/historical/powertrack/accounts/U
N/publishers/twitter/jobs/cw4zhby6s4.json"

# In[77]:

response_5 = requests.get(job_url,
                           auth=(os.getenv('FLOCK_USER'),
os.getenv('FLOCK_PASS'))))

# In[78]:

response_5_dict = json.loads(response_5.text)

# In[79]:
```





```
print(json.dumps(response_5_dict, indent=2))

# # 6. Get Completed Job Status
# In[2]:

job_url =
"https://api.flock.unglobalpulse.net/historical/powertrack/accounts/U
N/publishers/twitter/jobs/cw4zhby6s4.json"

# In[3]:

response_6 = requests.get(job_url,
                           auth=(os.getenv('FLOCK_USER'),
os.getenv('FLOCK_PASS'))))

# In[4]:

response_6_dict = json.loads(response_6.text)

# In[5]:

print(json.dumps(response_6_dict, indent=2))

# # 7. Get Completed Job Data Download URL List
# In[6]:

data_url = response_6_dict['results']['dataURL']

# In[7]:

print(data_url)

# In[8]:
```





```

response_7 = requests.get(data_url,
                           auth=(os.getenv('FLOCK_USER'),
os.getenv('FLOCK_PASS'))))

# In[9]:

response_7_dict = json.loads(response_7.text)

# In[10]:

print(json.dumps(response_7_dict, indent=2))

# # 8. Download Files

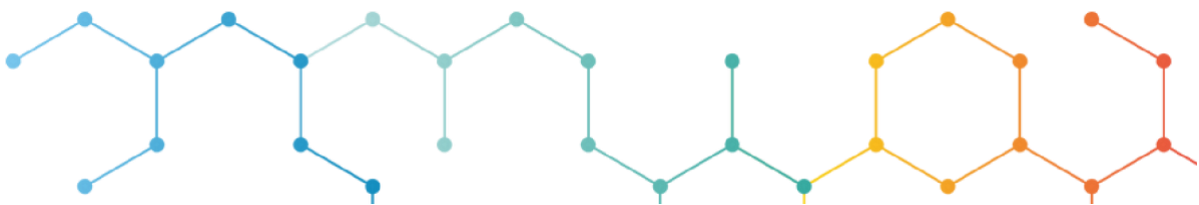
# In[11]:

from urllib.request import urlretrieve
from time import sleep
totalJsonGZFiles = len(response_7_dict["urlList"])
currentJsonGZFileCnt = 0
for oneUrl in response_7_dict["urlList"]:
    localFile = oneUrl.split('/')[-2] + '.json.gz'
    if localFile > '2': #'41577.json.gz':
        sleep(1)
        localFile = f"/enca/home/undp-saiz-
munoz/release2/results/{localFile}"
        print("Retrieving to {} ({} out of {})".format(localFile,
currentJsonGZFileCnt, totalJsonGZFiles))
        retrievedGood = False
        currentJsonGZFileCnt += 1
        while not retrievedGood:
            try:
                urlretrieve(oneUrl, localFile)
                retrievedGood = True
            except:
                waitSeconds = 10
                print("Error retrieving {}. Waiting {} seconds and
retrying...".format(localFile, waitSeconds))
                sleep(waitSeconds)

# # 9. Check file integrity and concatenate them all into one CSV
file

# In[12]:

```





```
import pandas as pd

# In[13]:

csv_columns = ["name","tag","created","text","retweetCount"]

# In[14]:

def write_to_csv_file(file_handle, one_result):
    i = 0
    df = pd.DataFrame(columns=csv_columns)

    long_body = None
    #location = None
    tag = None
    name = None
    try:
        long_body = one_result["long_object"]["body"]
    except:
        pass
    try:
        #location =
one_result["actor"]["location"]["displayName"]
        tag = one_result["gnip"]["matching_rules"][0]["tag"]
    except:
        pass
    try:
        name = one_result ["actor"]["displayName"]
    except:
        pass
    df.loc[i] =
[name,tag,one_result["postedTime"],one_result["body"],one_result["ret
weetCount"]]

    df.to_csv(file_handle, header=False, sep=',', encoding='utf-
8', columns=csv_columns, index=False)

# In[15]:

import gzip

# In[17]:
```





```
totalActivityCount = 0

with open("Analysis_August_22.csv", "w") as file_out_csv:
    file_out_csv.write(','.join(csv_columns)+"\n")

    for oneUrl in response_7_dict["urlList"]:
        localFile = oneUrl.split('/')[ -2] + '.json.gz'
        if localFile > '2' and localFile < '500000.json.gz':
# '41577.json.gz':
            localFile = f"/enca/home/undp-saiz-
munoz/release2/results/{localFile}"
            print("Reading {}".format(localFile))
            with gzip.open(localFile, 'rt', encoding='utf-8') as f:
                activityCount = 0
                expectedActivityCount = None
                for line in f:
                    line_dict = json.loads(line)
                    if "info" in line_dict.keys() and
"activity_count" in line_dict["info"].keys():
                        expectedActivityCount =
line_dict["info"]["activity_count"]
                    else:
                        activityCount += 1
                        write_to_csv_file(file_out_csv, line_dict)

                if activityCount != expectedActivityCount:
                    print("WARNING - activity count mismatch for {} -
expected {}, actual {}. You should re-download it.".
                        format(localFile, expectedActivityCount,
activityCount))
                totalActivityCount += activityCount

print("Total activity count: {}".format(totalActivityCount))

# ## Analysis

# In[72]:

import pandas as pd
df = pd.read_csv('Analysis_Total_July.csv')

# In[73]:

len(df)

# In[297]:
```





```
from SapGenderPrediction import GndrPrdct

Classifier = GndrPrdct()
def PredictGender(df_in):
    df = df_in
    genre_by_tweet = []
    for tweet in df['clean_text']:
        genre_by_tweet.append(Classifier.predict_gender(tweet))
    df["gender"] = genre_by_tweet

# In[298]:

import numpy as np
import re

def clean_tweet(tweet):
    stopwords = ["rt", "for", "on", "an", "a", "of", "and", "in",
                 "the", "to", "from", "girl",
                 "girls", "women", "woman", "s", "u", "t", "womens", "amp", "im", "m", "re"]

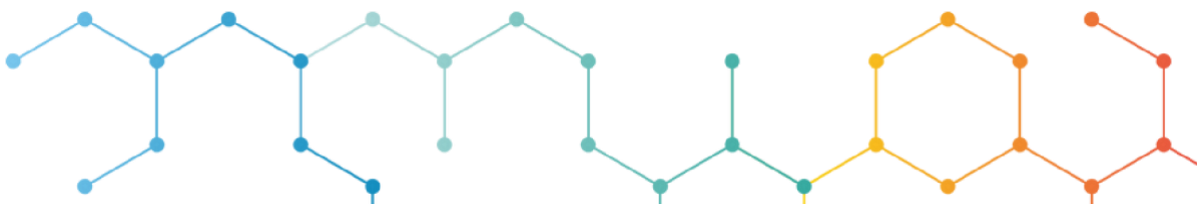
    if type(tweet) == np.float:
        return ""
    temp = tweet.lower()
    temp = re.sub("'", "", temp) # to avoid removing contractions in
    english
    temp = re.sub("@[A-Za-z0-9_]+", "", temp)
    temp = re.sub("#[A-Za-z0-9_]+", "", temp)
    temp = re.sub(r'http\S+', '', temp)
    temp = re.sub('[()!?!]', ' ', temp)
    temp = re.sub('\[.*?\]', ' ', temp)
    temp = re.sub("[^a-z0-9Ã€-Ã¿\u00f1\u00d1]", " ", temp)
    temp = temp.split()
    temp = [w for w in temp if not w in stopwords]
    temp = " ".join(word for word in temp)
    return temp

# In[299]:

df['clean_text'] = [clean_tweet(tw) for tw in df['text']]

# In[301]:

PredictGender(df)
```





```
# In[303]:
```

```
df_aggregated = df[['tag','created', 'retweetCount','clean_text',  
'gender']].copy()
```

```
# In[ ]:
```

```
df_aggregated.to_csv('Analysis_Total_July.csv')
```







Appendix B: Continuous data pipeline code

1.1 Phase I

```
# Databricks notebook source
#          MAGIC          %pip          install          --upgrade          -e
git+https://github.com/twintproject/twint.git@origin/master#egg=twint

# COMMAND -----

import nest_asyncio
nest_asyncio.apply()

# COMMAND -----


import twint
from time import sleep

# COMMAND -----

import pandas as pd

# COMMAND -----

scrapList = [{'Topic':"Education",'Country':"Uganda",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) AND geocode:0.347596,32.582520,200km)','Hate':0},
              {'Topic':"STEM",'Country':"Uganda",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) AND geocode:0.347596,32.582520,200km)','Hate':0},
              {'Topic':"Violence",'Country':"Uganda",'query':'((rape OR (sexual AND assault) OR (sexual AND violence) OR (sexual AND abuse) OR (forced AND sex) OR (child AND marriage) OR (sex AND trafficking) OR (child AND trafficking) OR (female AND genital AND mutilation) ) AND geocode:0.347596,32.582520,200km)','Hate':0},
```





{'Topic':"Reproduction",'Country':"Uganda",'query': '(abortion OR contraception OR (birth AND control) OR pill OR IUD OR (unwanted AND pregnancy)) AND geocode:0.347596,32.582520,200km)', 'Hate':0},

{'Topic':"Work",'Country':"Uganda",'query': '((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR care OR home OR domestic) AND geocode:0.347596,32.582520,200km)', 'Hate':0},

{'Topic':"Politics",'Country':"Uganda",'query': '((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends OR candidate) AND (lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR government OR politician OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresented) AND geocode:0.347596,32.582520,200km)', 'Hate':0},

{'Topic':"Education",'Country':"Colombia",'query': '((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (educación OR escuela OR escuelas OR colegio OR colegios OR educar OR educación OR universidad OR universidades OR uni OR maestro OR maestros OR profesor OR profesores OR aprendizaje OR curso OR enseñanza) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"STEM",'Country':"Colombia",'query': '((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (STEM OR ciencia OR código OR codificar OR coding OR hacker OR coder OR tecnología OR tech OR ingeniería OR matemáticas) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"Violence",'Country':"Colombia",'query': '((violación OR feminicidio OR (violencia AND machista) OR (violencia AND género) OR (agresión AND sexual) OR (abuso AND sexual) OR (abusos AND sexuales) OR (sexo AND sin AND consentimiento) OR (tráfico AND mujeres) OR (tráfico AND niños)) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"Reproduction",'Country':"Colombia",'query': '((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (aborto OR anticonceptivos OR (control AND natalidad) OR píldora OR DIU OR (embarazo AND no AND deseado)) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"Work",'Country':"Colombia",'query': '((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (trabajo OR trabajando OR empleo OR carrera OR oficina OR emplear OR empleado OR empleador OR ambición OR éxito OR fracaso OR promoción OR ascenso OR salario) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"Politics",'Country':"Colombia",'query': '((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias OR gobernadora OR candidata) AND (liderar OR líder OR líderes OR liderazgo OR poder OR poderoso OR política OR administración OR





administraciones OR gobierno OR gobiernos OR (partido AND político) OR corrupción OR voto) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"Education",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend) AND (edukasyon OR paaralan OR mga paaralan OR edu OR kolehiyo OR unibersidad OR guro OR pag-aaral OR kurso OR pagtuturo) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"STEM",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend) AND (STEM OR hacker OR science OR code OR coding OR teknolohiya OR engineering OR matematika OR tech) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Violence",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan) AND (panggagahasa OR sekswal na pag-atake OR sekswal na karahasan OR sekswal na pang-aabuso OR puwersahin ang sex OR child marriage OR kasal ng mga bata OR forced marriage OR sex trafficking OR child trafficking OR child trafficking) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Reproduction",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend) AND (pagpapalaglag OR pagpipigil sa pagbubuntis OR birth control OR tableta OR IUD OR hindi gustong pagbubuntis) AND geocode:geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Work",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend) AND (trabaho OR nagtatrabaho OR karera OR opisina OR ambisyon OR tagumpay OR kabiguan OR promosyon OR na-promote OR pagbabawas OR pagpapababa OR suweldo OR pagtaas OR pagbabayad OR anak OR bata OR sanggol OR pamilya OR tahanan OR domestic) AND geocode:geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Politics",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend OR kandidato) AND (pinuno OR pamumuno OR kapangyarihan OR makapangyarihan OR pulitika OR administrasyon OR (partidong AND pampulitika) OR gobyerno mga tiwali OR katiwalian OR bumoto OR mga boto OR pandaraya OR) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Education",'Country':"Philippines",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"STEM",'Country':"Philippines",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Violence",'Country':"Philippines",'query':'((rape OR (sexual AND assault) OR (sexual AND violence) OR (sexual AND abuse) OR (forced AND sex) OR (child AND marriage) OR (sex AND trafficking) OR (child AND trafficking)) AND geocode:14.599512,120.984222,200km)', 'Hate':0},





{'Topic':"Reproduction",'Country':"Philippines",'query': '(abortion OR contraception OR (birth AND control) OR pill OR IUD OR (unwanted AND pregnancy)) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Work",'Country':"Philippines",'query': '((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR care OR home OR domestic) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Politics",'Country':"Philippines",'query': '((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends OR candidate) AND (lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR government OR politician OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresented) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Education",'Country':"Uganda",'query': '((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"STEM",'Country':"Uganda",'query': '((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

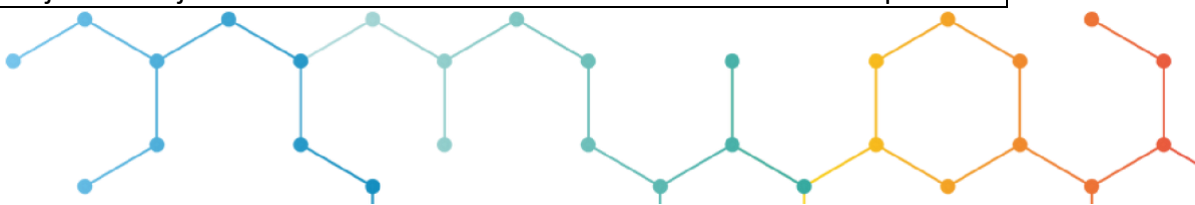
{'Topic':"Violence",'Country':"Uganda",'query': '((rape OR (sexual AND assault) OR (sexual AND violence) OR (sexual AND abuse) OR (forced AND sex) OR (child AND marriage) OR (sex AND trafficking) OR (child AND trafficking) OR (female AND genital AND mutilation)) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"Reproduction",'Country':"Uganda",'query': '(abortion OR contraception OR (birth AND control) OR pill OR IUD OR (unwanted AND pregnancy)) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"Work",'Country':"Uganda",'query': '((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR care OR home OR domestic) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"Politics",'Country':"Uganda",'query': '((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR government OR politician OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresented) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"Education",'Country':"Colombia",'query': '((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR





esposas OR novia OR novias) AND (educación OR escuela OR escuelas OR colegio OR colegios OR educar OR educación OR universidad OR universidades OR uni OR maestro OR maestros OR profesor OR profesores OR aprendizaje OR curso OR enseñanza) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"STEM",'Country':"Colombia",'query':'((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (STEM OR ciencia OR código OR codificar OR coding OR hacker OR coder OR tecnología OR tech OR ingeniería OR matemáticas) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"Violence",'Country':"Colombia",'query':'((violación OR feminicidio OR (violencia AND machista) OR (violencia AND género) OR (agresión AND sexual) OR (abuso AND sexual) OR (abusos AND sexuales) OR (sexo AND sin AND consentimiento) OR (tráfico AND mujeres) OR (tráfico AND niños)) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"Reproduction",'Country':"Colombia",'query':'((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (aborto OR anticonceptivos OR (control AND natalidad) OR píldora OR DIU OR (embarazo AND no AND deseado)) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

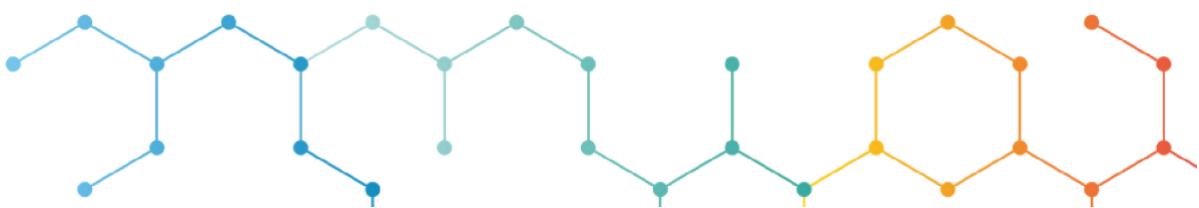
{'Topic':"Work",'Country':"Colombia",'query':'((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (trabajo OR trabajando OR empleo OR carrera OR oficina OR emplear OR empleado OR empleador OR ambición OR éxito OR fracaso OR promoción OR ascenso OR salario) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"Politics",'Country':"Colombia",'query':'((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias OR gobernadora OR candidata) AND (liderar OR líder OR líderes OR liderazgo OR poder OR poderoso OR política OR administración OR administraciones OR gobierno OR gobiernos OR (partido AND político) OR corrupción OR voto) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"Education",'Country':"Philippines",'query':'((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) AND geocode:14.599512,120.984222,200km)', 'Hate':1},

{'Topic':"STEM",'Country':"Philippines",'query':'((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) AND geocode:14.599512,120.984222,200km)', 'Hate':1},

{'Topic':"Violence",'Country':"Philippines",'query':'((rape OR (sexual AND assault) OR (sexual AND violence) OR (sexual AND abuse) OR (forced AND sex) OR (child AND marriage) OR (sex AND trafficking) OR (child AND trafficking)) AND geocode:14.599512,120.984222,200km)', 'Hate':1},



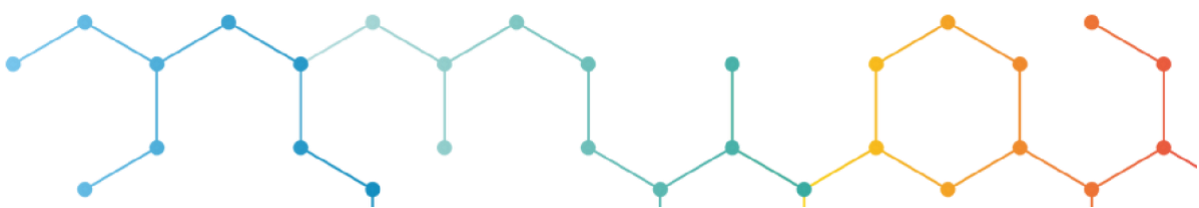


```
{'Topic':"Reproduction",'Country':"Philippines",'query': '(abortion OR
contraception OR (birth AND control) OR pill OR IUD OR (unwanted AND pregnancy)) AND
geocode:14.599512,120.984222,200km)', 'Hate':1},
{'Topic':"Work",'Country':"Philippines",'query': '((bimbo OR bitch OR cougar OR
crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR
whore) AND (work OR working OR career OR job OR employment OR office OR employ OR
employed OR employment OR ambition OR success OR failure OR promotion OR promoted
OR demotion OR demoted OR salary OR raise OR pay OR care OR home OR domestic ) AND
geocode:14.599512,120.984222,200km)', 'Hate':1},
{'Topic':"Politics",'Country':"Philippines",'query': '((bimbo OR bitch OR cougar
OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR
whore) AND (lead OR leader OR leaders OR leadership OR power OR powerful OR politics
OR administration OR government OR politician OR corrupt OR corruption OR vote OR
votes OR fraud OR misrepresented) AND
geocode:14.599512,120.984222,200km)', 'Hate':1}]

# COMMAND -----

df_final =
pd.DataFrame(columns=["date","username","tweet","nretweets","Country","Topic","Ha
te"])

for topic in range (42):
    config = twint.Config()
    config.Search = scrapList[topic]['query']
    config.Since = today
    config.Pandas = True
    twint.run.Search(config)
    df_twint = twint.storage.panda.Tweets_df
    print(scrapList[topic]['Topic'])
    if not df_twint.empty:
        df_twint["Country"] = scrapList[topic]['Country']
        df_twint["Topic"] = scrapList[topic]['Topic']
        df_twint["HateSpeech"] = scrapList[topic]['Hate']
```



1.2 Phase II

It should be noted that this implementation has posed a lot of problems due to the memory constraints of the Databricks cluster.

During the inference phase of the hate speech classification, all the inference steps have been parallellized thanks to the use of the huggingface arrow technology, powered by spark.

Arrow enables large amounts of data to be processed and moved quickly. It is a specific data format that stores data in a columnar memory layout. This provides several significant advantages:

- Arrow's standard format allows zero-copy reads which removes virtually all serialization overhead.
- Arrow is language-agnostic so it supports different programming languages.
- Arrow is column-oriented so it is faster at querying and processing slices or columns of data.
- Arrow allows for copy-free hand-offs to standard machine learning tools such as NumPy, Pandas, PyTorch, and TensorFlow.
- Arrow supports many, possibly nested, column types.

It shall be noted that during the tokenization step, the inference phase does not have any mechanism to map the tokens to the dataset in parallel, hence serialisation has been used instead. This results in a bottleneck that for big number of tweets ($\sim >1500$) leads to a memory exception inside the cluster:



To overcome this, several approaches have been tested:

- Import the multilingual hate speech model to spark NLP, that implements the tokenization step inside its own parallel pipeline. This did not work because Spark NLP does not yet support Roberta fine-tuned on specific tasks (see XXX)
- Use huggingface map for multiprocessing (<https://huggingface.co/docs/datasets/process>). In this case, the huggingface implementation did not recognize the hardware architecture of our Databricks cluster.
- Implement the split technique used in huggingface for streaming data as a preprocessing step prior to creating the huggingface dataset. This brute force approach gave good results and permits to scale up or down our system (including additional countries, extending the taxonomy, etc..) without having to deal with memory issues.

It shall be noted as well that the processing time of the pipeline using huggingface datasets has been decreased from a mean of ~40min/day using Spark NLP to ~25min/day.



```
#!/usr/bin/env python
# coding: utf-8

# In[ ]:

get_ipython().run_line_magic('pip', 'install --upgrade -e
git+https://github.com/twintproject/twint.git@origin/master#egg=twint')
get_ipython().system('pip install datasets')

# In[ ]:

import nest_asyncio
nest_asyncio.apply()

# In[ ]:

import twint
from time import sleep

# In[ ]:

import pandas as pd

# In[ ]:

from pyspark.sql.types import StructType, StructField, StringType, IntegerType
from pyspark.sql.functions import col
import pyspark.sql.functions as F
from pyspark.sql.functions import regexp_replace

# In[ ]:

import numpy as np
```



```
# In[ ]:
```

```
from datasets import load_dataset, Dataset, DatasetDict
import transformers
from transformers import AutoTokenizer
from transformers import BertTokenizerFast
from transformers import TFBertModel
from transformers import RobertaTokenizerFast
from transformers import TFRobertaModel
from transformers import AutoModel
from transformers import AutoModelForSequenceClassification
import tensorflow as tf
```

```
# In[ ]:
```

```
#####
```

```
import re
import html.entities
def lmap(f,xs):
    return list(map(f,xs))
```

```
#####
```

```
# The following strings are components in the regular expression
# that is used for tokenizing. It's important that phone_number
# appears first in the final regex (since it can contain whitespace).
# It also could matter that tags comes after emoticons, due to the
# possibility of having text like
#
# <:| and some text >:)
#
# Most importantly, the final element should always be last, since it
# does a last ditch whitespace-based tokenization of whatever is left.
```

```
# This particular element is used in a couple ways, so we define it
# with a name:
```

```
emoticon_string = r"""
(?:
  [<>]?
  [;:=8]          # eyes
  [\-o\*\']?      # optional nose
  [\\]\[\[dDpP/\:\]\{@\|\\] # mouth
  |
  [\\]\[\[dDpP/\:\]\{@\|\\] # mouth
  [\-o\*\']?      # optional nose
```



```

        [;=8]          # eyes
        [<>]?
    )""

# The components of the tokenizer:
regex_strings = (
    # Phone numbers:
    r"""
    (?
    (?      # (international)
        \+?[01]
        [\-\s.]*
    )?
    (?      # (area code)
        [\(\]?
        \d{3}
        [\-\s.\\]*
    )?
    \d{3}    # exchange
    [\-\s.]*
    \d{4}    # base
    )""
    ,
    # Emoticons:
    emoticon_string
    ,
    # HTML tags:
    r"""<[^>]+>"""
    ,
    # Twitter username:
    r"""(?:@[\w_]+)"""
    ,
    # Twitter hashtags:
    r"""(?:#[\w_]+[\w'_-]*[\w_]+)"""
    ,
    # Remaining word types:
    r"""
    (?:[a-z][a-z'\-_]+[a-z])    # Words with apostrophes or dashes.
    |
    (?:[+-]?\d+[/\.-]\d+[+-]?) # Numbers, including fractions, decimals.
    |
    (?:[\w_]+)                  # Words without apostrophes or dashes.
    |
    (?:\.(?:\s*\.){1,})        # Ellipsis dots.
    |
    (?:\S)                      # Everything else that isn't whitespace.
    """

```



```

)

#####
# This is the core tokenizing regex:

word_re = re.compile(r""""(%s)"""" % "|".join(regex_strings), re.VERBOSE | re.I |
re.UNICODE)

# The emoticon string gets its own regex so that we can preserve case for them as
needed:
emoticon_re = re.compile(regex_strings[1], re.VERBOSE | re.I | re.UNICODE)

# These are for regularizing HTML entities to Unicode:
html_entity_digit_re = re.compile(r"&\#\d+;")
html_entity_alpha_re = re.compile(r"&\w+;")
amp = "&"

#####

class Tokenizer:
    def __init__(self, preserve_case=False):
        self.preserve_case = preserve_case

    def tokenize(self, s):
        """
        Argument: s -- any string or unicode object
        Value: a tokenize list of strings; conatenating this list returns the original string if
preserve_case=False
        """
        # Try to ensure unicode:
        #try:
        #s = str(s,'utf-8')
        #except UnicodeDecodeError:
        #s = str(s).encode('string_escape')
        #s = str(s,'utf-8')
        # Fix HTML character entitites:
        s = self.__html2unicode(s)
        # Tokenize:
        words = word_re.findall(s)
        # Possible alter the case, but avoid changing emoticons like :D into :d:
        if not self.preserve_case:
            words = lmap((lambda x : x if emoticon_re.search(x) else x.lower()), words)
        return words

    def __html2unicode(self, s):
        """
        Internal metod that seeks to replace all the HTML entities in

```




```

s with their corresponding unicode characters.
"""

# First the digits:
ents = set(html_entity_digit_re.findall(s))
if len(ents) > 0:
    for ent in ents:
        entnum = ent[2:-1]
        try:
            entnum = int(entnum)
            s = s.replace(ent, unichr(entnum))
        except:
            pass
# Now the alpha versions:
ents = set(html_entity_alpha_re.findall(s))
ents = filter((lambda x : x != amp), ents)
for ent in ents:
    entname = ent[1:-1]
    try:
        s = s.replace(ent, unichr(htmlentitydefs.name2codepoint[entname]))
    except:
        pass
    s = s.replace(amp, " and ")
return s

#####
#####

#if __name__ == '__main__':
#tok = Tokenizer(preserve_case=False)
#samples = (
#u"RT @ #happyfuncoding: this is a typical Twitter tweet :-)",
#u"HTML entities & amp; other Web oddities can be an &acute;cute <em
class='grumpy'>pain</em> >:",
#u"It's perhaps noteworthy that phone numbers like +1 (800) 123-4567, (800) 123-
4567, and 123-4567 are treated as words despite their whitespace."
#)

#for s in samples:
#print
#=====
#print s
#tokenized = tok.tokenize(s)
#print "\n".join(tokenized)

# In[ ]:

```



```

import csv
from math import sin

class GndrPrdct:
    """Takes text and provides gender prediction (1 is female, 0 is male)"""
    def
__init__(self,fp="/dbfs/FileStore/shared_uploads/maria.saiz.munoz@undp.org/gender_le
x.csv"):
    self.tknzr = Tokenizer(preserve_case=False)
    self.weights = dict()
    with open(fp) as f:
        rdr = csv.reader(f)
        rdr.__next__()
        for r in rdr:
            self.weights[r[0]]=float(r[1])

    def weigh(self,token,tokens):
        w = self.weights.get(token,0)
        if w == 0:
            return 0
        else:
            return w*tokens.count(token)/len(tokens)

    def predict_gender(self,txt):
        tkns = list(self.tknzr.tokenize(txt))
        wts = sum([self.weigh(t,tkns) for t in set(tkns)])
        p = sin(-0.06724152+wts)
        if p >= 0:
            # Female
            return 1
        else:
            # Male
            return 0

# In[ ]:

import re

def clean_tweet(tweet):
    if type(tweet) == np.float:
        return ""
    if type(tweet) == np.int:
        return ""
    temp = tweet.lower()

```



```

temp = re.sub("", "", temp)
temp = re.sub("@[A-Za-z0-9_]+", "", temp)
temp = re.sub("#[A-Za-z0-9_]+", "", temp)
temp = re.sub(r'http\S+', "", temp)
temp = re.sub('([!?\']', '', temp)
temp = re.sub('\[.*?\]', '', temp)
temp = re.sub("[^a-z0-9Ã-ÿ\u00f1\u00d1]", " ", temp)
temp = temp.split()
temp = " ".join(word for word in temp)
if temp == "":
    temp = "Empty string"
temp = str(temp)
return temp

```

In[]:

```

from datetime import datetime, timedelta
currentMonth = datetime.now().month
currentYear = datetime.now().year
today = (datetime.today() - timedelta(days=7)).strftime('%Y-%m-%d')

```

In[]:

```

Classifier = GndrPrdct()
def PredictGender(df_in):
    df = df_in
    genre_by_tweet = []
    for tweet in df['tweet']:
        genre_by_tweet.append(Classifier.predict_gender(tweet))
    df["gender"] = genre_by_tweet
    return df

```

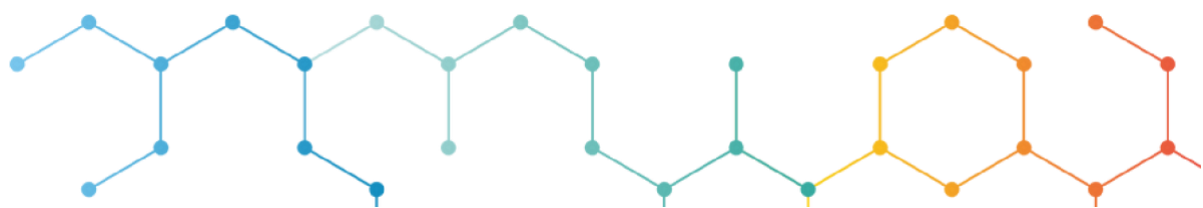
In[]:

```

tokenizer = AutoTokenizer.from_pretrained("xlm-roberta-base")
model =
AutoModelForSequenceClassification.from_pretrained("/dbfs/FileStore/multi_hatespeech_model")

```

In[]:



```

def predictDataset(dataset):
    def prediction(batch):
        encoded_input = tokenizer(batch["tweet"],padding=True,truncation=True,
return_tensors='pt')
        output = model(**encoded_input)
        return output

    hatespeech_encoded = dataset.map(prediction,batched=True,batch_size=None)

    return hatespeech_encoded

```

In[]:

```

def NormalizeTransformerOutput(output):
    predictions = tf.math.softmax(output_pred["train"]["logits"], axis=-1)
    def discretize(preds):
        result = 0
        if(preds[0] > 0.5):
            result = 0
        else:
            result = 1
        return result

    return [discretize(tw) for tw in predictions]

```

In[]:

```

scrapList = [{'Topic':"Education",'Country':"Uganda",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) AND geocode:0.347596,32.582520,200km)','Hate':0},
              {'Topic':"STEM",'Country':"Uganda",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) AND geocode:0.347596,32.582520,200km)','Hate':0},
              {'Topic':"Violence",'Country':"Uganda",'query':'((rape OR (sexual AND assault) OR (sexual AND violence) OR (sexual AND abuse) OR (forced AND sex) OR (child AND marriage) OR (sex AND trafficking) OR (child AND trafficking) OR (female AND genital AND mutilation) ) AND geocode:0.347596,32.582520,200km)','Hate':0},

```



{'Topic':"Reproduction",'Country':"Uganda",'query': '(abortion OR contraception OR (birth AND control) OR pill OR IUD OR (unwanted AND pregnancy)) AND geocode:0.347596,32.582520,200km)', 'Hate':0},

{'Topic':"Work",'Country':"Uganda",'query': '(((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR care OR home OR domestic) AND geocode:0.347596,32.582520,200km)', 'Hate':0},

{'Topic':"Politics",'Country':"Uganda",'query': '(((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends OR candidate) AND (lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR government OR politician OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresented) AND geocode:0.347596,32.582520,200km)', 'Hate':0},

{'Topic':"Education",'Country':"Colombia",'query': '(((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (educación OR escuela OR escuelas OR colegio OR colegios OR educar OR educación OR universidad OR universidades OR uni OR maestro OR maestros OR profesor OR profesores OR aprendizaje OR curso OR enseñanza) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

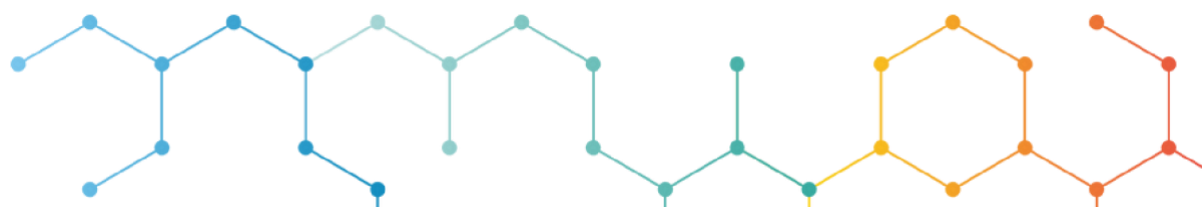
{'Topic':"STEM",'Country':"Colombia",'query': '(((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (STEM OR ciencia OR código OR codificar OR coding OR hacker OR coder OR tecnología OR tech OR ingeniería OR matemáticas) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"Violence",'Country':"Colombia",'query': '(((violación OR feminicidio OR (violencia AND machista) OR (violencia AND género) OR (agresión AND sexual) OR (abuso AND sexual) OR (abusos AND sexuales) OR (sexo AND sin AND consentimiento) OR (tráfico AND mujeres) OR (tráfico AND niños)) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"Reproduction",'Country':"Colombia",'query': '(((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (aborto OR anticonceptivos OR (control AND natalidad) OR píldora OR DIU OR (embarazo AND no AND deseado)) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"Work",'Country':"Colombia",'query': '(((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (trabajo OR trabajando OR empleo OR carrera OR oficina OR emplear OR empleado OR empleador OR ambición OR éxito OR fracaso OR promoción OR ascenso OR salario) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},

{'Topic':"Politics",'Country':"Colombia",'query': '(((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias OR gobernadora OR candidata) AND (liderar OR líder OR líderes OR liderazgo OR poder OR poderoso OR política OR administración OR administraciones OR gobierno OR gobiernos OR (partido AND político) OR corrupción OR voto) AND geocode:4.624335,-74.063644,200km)', 'Hate':0},



{'Topic':"Education",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend) AND (edukasyon OR paaralan OR mga paaralan OR edu OR kolehiyo OR unibersidad OR guro OR pag-aaral OR kurso OR pagtuturo) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"STEM",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend) AND (STEM OR hacker OR science OR code OR coding OR teknolohiya OR engineering OR matematika OR tech) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Violence",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan) AND (panggagahasa OR sekswal na pag-atake OR sekswal na karahasan OR sekswal na pang-aabuso OR puwersahin ang sex OR child marriage OR kasal ng mga bata OR forced marriage OR sex trafficking OR child trafficking OR child trafficking) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Reproduction",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend) AND (pagpapalaglag OR pagpipigil sa pagbubuntis OR birth control OR tableta OR IUD OR hindi gustong pagbubuntis) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Work",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend) AND (trabaho OR nagtatrabaho OR karera OR opisina OR ambisyon OR tagumpay OR kabiguan OR promosyon OR na-promote OR pagbabawas OR pagpapababa OR suweldo OR pagtaas OR pagbabayad OR anak OR bata OR sanggol OR pamilya OR tahanan OR domestic) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Politics",'Country':"Philippines",'query':'((babae OR balo OR biyuda OR ina OR asawa OR kasintahan OR girlfriend OR kandidato) AND (pinuno OR pamumuno OR kapangyarihan OR makapangyarihan OR pulitika OR administrasyon OR (partidong AND pampulitika) OR gobyerno mga tiwali OR katiwalian OR bumoto OR mga boto OR pandaraya OR) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Education",'Country':"Philippines",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"STEM",'Country':"Philippines",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends) AND (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Violence",'Country':"Philippines",'query':'((rape OR (sexual AND assault) OR (sexual AND violence) OR (sexual AND abuse) OR (forced AND sex) OR (child AND marriage) OR (sex AND trafficking) OR (child AND trafficking)) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Reproduction",'Country':"Philippines",'query':'(abortion OR contraception OR (birth AND control) OR pill OR IUD OR (unwanted AND pregnancy)) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Work",'Country':"Philippines",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives



OR girlfriend OR girlfriends) AND (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR care OR home OR domestic) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Politics",'Country':"Philippines",'query':'((woman OR women OR girl OR girls OR female OR females OR widow OR widows OR mother OR mothers OR wife OR wives OR girlfriend OR girlfriends OR candidate) AND (lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR government OR politician OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresented) AND geocode:14.599512,120.984222,200km)', 'Hate':0},

{'Topic':"Education",'Country':"Uganda",'query':'((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"STEM",'Country':"Uganda",'query':'((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"Violence",'Country':"Uganda",'query':'((rape OR (sexual AND assault) OR (sexual AND violence) OR (sexual AND abuse) OR (forced AND sex) OR (child AND marriage) OR (sex AND trafficking) OR (child AND trafficking) OR (female AND genital AND mutilation)) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"Reproduction",'Country':"Uganda",'query':'(abortion OR contraception OR (birth AND control) OR pill OR IUD OR (unwanted AND pregnancy)) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"Work",'Country':"Uganda",'query':' ((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR care OR home OR domestic) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"Politics",'Country':"Uganda",'query':'((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (lead OR leader OR leaders OR leadership OR power OR powerful OR politics OR administration OR government OR politician OR corrupt OR corruption OR vote OR votes OR fraud OR misrepresented) AND geocode:0.347596,32.582520,200km)', 'Hate':1},

{'Topic':"Education",'Country':"Colombia",'query':'((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (educación OR escuela OR escuelas OR colegio OR colegios OR educar OR educación OR universidad OR universidades OR uni OR maestro OR maestros OR profesor OR profesores OR aprendizaje OR curso OR enseñanza) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"STEM",'Country':"Colombia",'query':'((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (STEM OR ciencia OR código OR codificar OR coding OR



hacker OR coder OR tecnología OR tech OR ingeniería OR matemáticas) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"Violence",'Country':"Colombia",'query':'((violación OR feminicidio OR (violencia AND machista) OR (violencia AND género) OR (agresión AND sexual) OR (abuso AND sexual) OR (abusos AND sexuales) OR (sexo AND sin AND consentimiento) OR (tráfico AND mujeres) OR (tráfico AND niños)) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"Reproduction",'Country':"Colombia",'query':'((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (aborto OR anticonceptivos OR (control AND natalidad) OR píldora OR DIU OR (embarazo AND no AND deseado)) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"Work",'Country':"Colombia",'query':'((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias) AND (trabajo OR trabajando OR empleo OR carrera OR oficina OR emplear OR empleado OR empleador OR ambición OR éxito OR fracaso OR promoción OR ascenso OR salario) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"Politics",'Country':"Colombia",'query':'((mujer OR mujeres OR niña OR niñas OR mujer OR mujeres OR viuda OR viudas OR madre OR madres OR esposa OR esposas OR novia OR novias OR gobernadora OR candidata) AND (liderar OR líder OR líderes OR liderazgo OR poder OR poderoso OR política OR administración OR administraciones OR gobierno OR gobiernos OR (partido AND político) OR corrupción OR voto) AND geocode:4.624335,-74.063644,200km)', 'Hate':1},

{'Topic':"Education",'Country':"Philippines",'query':'((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (education OR school OR schools OR educate OR edu OR college OR uni OR university OR teacher OR teachers OR learning OR course OR teaching) AND geocode:14.599512,120.984222,200km)', 'Hate':1},

{'Topic':"STEM",'Country':"Philippines",'query':'((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (STEM OR hacker OR science OR code OR coding OR technology OR engineering OR mathematics OR tech) AND geocode:14.599512,120.984222,200km)', 'Hate':1},

{'Topic':"Violence",'Country':"Philippines",'query':'((rape OR (sexual AND assault) OR (sexual AND violence) OR (sexual AND abuse) OR (forced AND sex) OR (child AND marriage) OR (sex AND trafficking) OR (child AND trafficking)) AND geocode:14.599512,120.984222,200km)', 'Hate':1},

{'Topic':"Reproduction",'Country':"Philippines",'query':'(abortion OR contraception OR (birth AND control) OR pill OR IUD OR (unwanted AND pregnancy)) AND geocode:14.599512,120.984222,200km)', 'Hate':1},

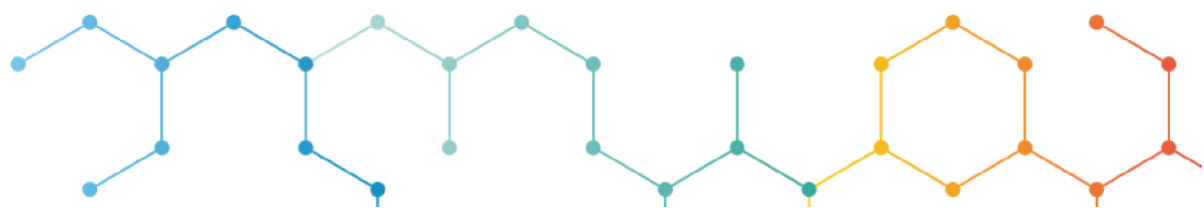
{'Topic':"Work",'Country':"Philippines",'query':'((bimbo OR bitch OR cougar OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR whore) AND (work OR working OR career OR job OR employment OR office OR employ OR employed OR employment OR ambition OR success OR failure OR promotion OR promoted OR demotion OR demoted OR salary OR raise OR pay OR care OR home OR domestic) AND geocode:14.599512,120.984222,200km)', 'Hate':1},




```
{'Topic':'Politics','Country':'Philippines','query':'((bimbo OR bitch OR cougar
OR crone OR cunt OR old digger OR hag OR slut OR spinster OR squaw OR twat OR wag OR
whore) AND (lead OR leader OR leaders OR leadership OR power OR powerful OR politics
OR administration OR government OR politician OR corrupt OR corruption OR vote OR
votes OR fraud OR misrepresented) AND
geocode:14.599512,120.984222,200km)','Hate':1]}
```

```
# In[ ]:
```

```
df_final =
pd.DataFrame(columns=["date","username","tweet","Country","Topic","HateSpeech"])
for topic in range (len(scrapList)):
    config = twint.Config()
    config.Search = scrapList[topic]['query']
    config.Since = today
    config.Pandas = True
    config.Hide_output = True
    twint.run.Search(config)
    df_twint = twint.storage.panda.Tweets_df
    print(scrapList[topic]['Topic'])
    print(scrapList[topic]['Country'])
    print("SIZE TWEETS:")
    print(len(df_twint))
    if not df_twint.empty:
        slices = (len(df_twint)/1000) + 1
        slice = 0
        while(slice < slices):
            if(len(df_twint) < 1000):
                df_process = df_twint
            else:
                min_val = 1000*(slice)
                if (slice == (slices-1)):
                    max_val = (len(df_twint)-1)
                else:
                    max_val = (1000*(slice+1))
                print("min_val:")
                print(min_val)
                print("max_val:")
                print(max_val)
                df_process = df_twint[min_val:max_val]
                df_process["Country"] = scrapList[topic]['Country']
                df_process["Topic"] = scrapList[topic]['Topic']
                df_process["HateSpeech"] = scrapList[topic]['Hate']
                df_process['tweet'] = [clean_tweet(tw) for tw in df_process['tweet']]
            # Gender classification
```



```

df_process = PredictGender(df_process)
#Hate Speech classification
df_process = df_process.dropna(subset=['tweet'])
res = df_process[df_process["HateSpeech"] == 0]
res_hate = df_process[df_process["HateSpeech"] == 1]
df_dataset = res[['tweet','date']]
df_dataset.to_csv("hs_dataset.csv",index=False)
twitter = load_dataset("csv",data_files="hs_dataset.csv")
output_pred = predictDataset(twitter)
if("logits" in output_pred["train"].column_names):
    predictions = tf.math.softmax(output_pred["train"]["logits"], axis=-1)
    res["HateSpeech"] = NormalizeTransformerOutput(predictions)
df_process = pd.concat([res,res_hate])
frames = [df_final,
df_process[["date","username","tweet","nretweets","Country","Topic","HateSpeech","gender"]]]

df_final = pd.concat(frames)
slice = slice + 1

```

In[]:

```

df_final = df_final.rename(columns = {'tweet':'text'})
df_final

```

Topic modelling

In[]:

```

df_final["HateSpeech"] = df_final["HateSpeech"].astype(int)
df_final["gender"] = df_final["gender"].astype(int)
df_pandas = df_final

```

In[]:

```

df_pandas['Topic'].replace(to_replace=["Education","STEM","Violence","Reproduction",
"Work","Politics"], value=[1, 2,3,4,5,6], inplace=True)

```

In[]:



```

from transformers import pipeline
import pandas as pd

df_hate = df_pandas[df_pandas["HateSpeech"]==1]
df_hate["Subtopic"] = ""

subtopics = [
    ["Policy", "Career", "Money", "Diversity", "Other"],
    ["Carrer", "Business", "Scholarships", "Coding", "Other"],
    ["Sexual Violence", "Racism", "Laws", "Crime", "Other"],
    ["Abortion", "HIV", "Crime", "LGBT", "Other"],
    ["Wealth", "Gender inequality", "Stereotypes", "Success", "Other"],
    ["Emigration", "Public finance", "Leadership", "Violence", "Other"]
]

classifier = pipeline("zero-shot-classification")

for index, row in df_hate.iterrows():
    sequence = row["text"]

    output = classifier(sequence, subtopics[row["Topic"]-1])
    if(output["scores"][0]>=0.5):
        df_hate.at[index, "Subtopic"] = output["labels"][0]
    else:
        df_hate.at[index, "Subtopic"] = "Other"

# In [ ]:

df_nonhate = df_pandas[df_pandas["HateSpeech"]=="0"]
df_nonhate["Subtopic"] = ""

# In [ ]:

frames = [df_nonhate, df_hate]

result = pd.concat(frames)

# In [ ]:

result['Subtopic'].replace(to_replace=["", "Policy", "Career", "Money", "Diversity", "Other"], value=[0,1, 2,3,4,5], inplace=True)
result['Subtopic'].replace(to_replace=["", "Carrer", "Business", "Scholarships", "Coding", "Other"], value=[0,1, 2,3,4,5], inplace=True)

```



```

    result['Subtopic'].replace(to_replace=["", "Sexual Violence", "Racism",
    "Laws", "Crime", "Other"], value=[0,1, 2,3,4,5], inplace=True)
    result['Subtopic'].replace(to_replace=["", "Abortion", "HIV", "Crime", "LGBT", "Other"],
    value=[0,1, 2,3,4,5], inplace=True)
    result['Subtopic'].replace(to_replace=["", "Wealth", "Gender
    inequality", "Stereotypes", "Success", "Other"], value=[0,1, 2,3,4,5], inplace=True)
    result['Subtopic'].replace(to_replace=["", "Emigration", "Public
    finance", "Leadership", "Violence", "Other"], value=[0,1, 2,3,4,5], inplace=True)

```

```
# In[ ]:
```

```
result
```

```
# ## Transform into json and send to server
```

```
# In[ ]:
```

```
def GetMale(gen):
```

```
    if gen == 0:
```

```
        return 1
```

```
    else:
```

```
        return 0
```

```
def GetHateGender(sum):
```

```
    if sum == 2:
```

```
        return 1
```

```
    else:
```

```
        return 0
```

```
def sendjson(df, fileName):
```

```
    ftp = FTP('gnldm1026.siteground.biz')
```

```
    ftp.login("mariasaimunoz@data.undp.org", "MibebeGuille_21")
```

```
    ftp.cwd('data.undp.org/public_html/EWS')
```

```
    #ftp.cwd('staging2.data.undp.org/EWS')
```

```
    df.to_json(fileName, orient='records', indent=4)
```

```
    with open(fileName, "rb") as f:
```

```
        ftp.storbinary('STOR ' + os.path.basename(fileName), f)
```

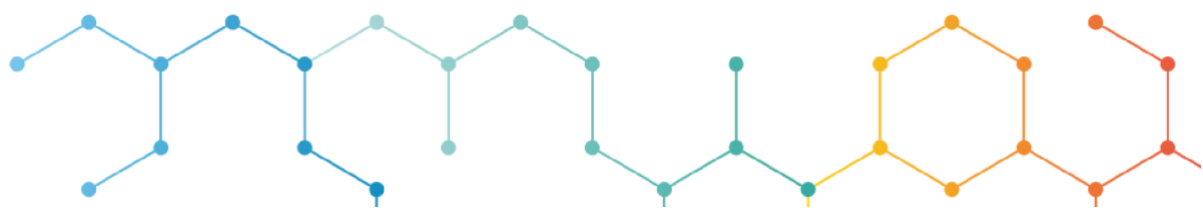
```
    ftp.quit()
```

```
def sendcsv(df, fileName):
```

```
    ftp = FTP('gnldm1026.siteground.biz')
```

```
    ftp.login("mariasaimunoz@data.undp.org", "MibebeGuille_21")
```

```
    ftp.cwd('data.undp.org/public_html/EWS')
```



```

ftp.cwd('staging2.data.undp.org/EWS')
df.to_csv(fileName,index=False)
with open(fileName, "rb") as f:
    ftp.storbinary('STOR ' + os.path.basename(fileName), f)
ftp.quit()

def TruncateOutlier(value,maxi):
    if(value> maxi):
        return (maxi+ np.random.choice([-15,15]))
    else:
        return value

def TruncateOutlier2(value,maxi):
    if(value> maxi):
        return (int(round(maxi/2)) + np.random.choice([-5,5]))
    else:
        return value

#Formato de entrada: Date | Hour | Tweet | nretweets | Country | Topic | HateSpeech
| gender
def SendjsonFile(df,fileName):
    df = df.drop("tweet",axis=1)

    df["tweets"] = 1

    df["Male"] = [GetMale(tw) for tw in df['gender']]
    df["Female"] = df["gender"]
    df["Male"] = df["Male"].astype(int)
    df["Female"] = df["Female"].astype(int)
    df["HateSpeech"] = df["HateSpeech"].fillna(0)
    df["HateSpeech"] = df["HateSpeech"].astype(int)
    df["temp"] = df["Male"] + df["HateSpeech"]

    df["MaleHate"] = [GetHateGender(tw) for tw in df['temp']]
    df = df.drop("temp",axis=1)
    df["temp"] = df["Female"] + df["HateSpeech"]
    df["FemaleHate"]=[GetHateGender(tw) for tw in df['temp']]
    df = df.drop("temp",axis=1)

    f = {'tweets': 'sum', 'Male': 'sum', 'MaleHate': 'sum', 'FemaleHate': 'sum'}
    df_json1 = df.groupby(["Hour", "date","Topic"], as_index=False).agg(f)
    upper_limit = (df_json1["tweets"].median()*100)
    df_json1["tweets"] = [TruncateOutlier(tw,upper_limit) for tw in df_json1['tweets']]
    df_json1["Male"] = [TruncateOutlier2(tw,upper_limit) for tw in df_json1['Male']]

    f = {'tweets': 'sum', 'Male': 'sum','MaleHate': 'sum', 'FemaleHate': 'sum'}
    df_json2 = df_json1.groupby(["Hour", "date"], as_index=False).agg(f)

```



```

df_json2['Topic'] = 0

out = pd.concat([df_json1,df_json2])
out = out.sort_values(by=['date'])
df['tweets','Male','MaleHate','FemaleHate'] =
df['tweets','Male','MaleHate','FemaleHate'].apply(lambda x: x*2)
strFile= "/dbfs/FileStore/shared_uploads/maria.saiz.munoz@undp.org/"+fileName

df = pd.read_csv(strFile)

output = pd.concat([df,out])

f = {'tweets': 'sum', 'Male': 'sum', 'MaleHate': 'sum', 'FemaleHate': 'sum'}
output = output.groupby(["Hour", "date", "Topic"], as_index=False).agg(f)

output.to_csv(strFile)

sendcsv(output,fileName)

# In[ ]:

from ftplib import FTP
import os

result['date'] = pd.to_datetime(result['date'], format="%Y-%m-%d %H:%M:%f",
errors='coerce')
result["Hour"] = [tw.hour for tw in result['date']]
result = result[result.date.notnull()]
result['date'] = result['date'].dt.strftime('%m-%d-%Y')
result['Country'].replace(to_replace=["Uganda","Colombia","Philippines"],
value=[1,2,3], inplace=True)
result = result.drop("text",axis=1)

df_result = result[result['Country']==1]
fileName = "UgandaData.csv"
SendjsonFile(df_result,fileName)

df_result = result[result['Country']==2]
fileName = "ColombiaData.csv"
SendjsonFile(df_result,fileName)

df_result = result[result['Country']==3]
fileName = "PhilippinesData.csv"

```





SendjsonFile(df_result,fileName)





Appendix C: Concept note AI EWS

Gender Team Global Observatory of Gender Responsive Policies
DRAFT AI Early Warning System Concept Note
And AI for COVID-19 Global Gender Response Tracker updates

Background

A Global Observatory of Public Policies is being set up by UNDP to develop gender responsive policy measures to achieve gender equality. The observatory will carry out collaborative research and data analytics, identify good practices linked to laws and public policies and promote policy dialogue. The Observatory will capture policy measures taken by governments on economic security – labor market, fiscal and economic measures and social security; and gender parity in decision making in public administration. The Observatory will provide guidance for policymakers and evidence for advocates to ensure a gender-sensitive policy response. The Global Observatory will be anchored in the successful work of the UNDP and UNW COVID Global Gender Policy Tracker.

The Early Warning System on gender backlashes: What is it?

A Global platform and Early Warning System will be designed and set up with an AI software company to develop the early-warning system to scan social networks, social media and mass media for trends and signs of specific gender related events, tracking public discussions and public opinions for indications of backlashes relating to gender equality and women's rights.

We would like to understand how we can use AI, machine learning and big data to understand social changes, people's opinions and track backlashes on gender equality.

This project will analyse social media, and language, to understand social narratives, complaints and concerns around policies and government decisions. This would enable us to build an early warning system through developing indicators of where there is an increase in areas of concern or complaint relating and monitoring public opinions relating to gender equality. These would give signals of increasing areas of risk. The early warning system will track any backlashes on gender equality to enable policy responses.

Piloting and scale up: Country Coverage

Initially we would seek to pilot the AI early warning system with a selected number of countries (eg 5 across our 5 regions). These countries would be selected in consultation with regional gender advisers, based on criteria including

- capacity to participate in the pilot
- capacity to use the data
- availability of data in the country

If successful, this 5-country pilot package would then be scaled up and rolled out to other countries.





Data parameters

- Data would ideally be collected on a daily basis to provide real time data (rather than snapshot of data at specific times)
- Ideally data would be available at national and subnational level.
- The social narratives would need to monitor across languages. – perhaps 3 to begin with (eg English, French, Arabic ?)

Public platform

A public website would be set up to enable access to the data. This would enable stakeholders including CSOs and women's organisations to have access to the data, to understand and spotlight where gender backlashes occur and develop programme and policy responses, holding governments to account

The Gender team would like to maintain ownership over the AI early warning system and house it under the umbrella of the Global Observatory of Gender Responsive Policies, alongside the COVID 19 Global Gender Response Tracker, as well as the Public Sector Seal platform and the Gender Norms index. The Gender Team would lead the design, methodology, categories of analysis, visualisations, and branding of the platform product. We would like to understand how this will link to the Data Futures Platform whilst maintaining a distinct Gender Team identity.

A key principle for us is to take an ethical approach. We need to understand how the company will ensure an ethical and inclusive approach to the analysis of social media and big data, ensuring that privacy is a priority, using anonymized data. We need to understand how data providers treat their users and their data in an ethical way. We want to understand how to ensure that the data is inclusive.

Topics


Topics for tracking with AI on social media, to represent dominant perspectives and popular dialogue regarding gender equality and women's empowerment in public discourse, across countries, regions.

Considerations to keep in mind:

- 1.To try and get a sense of whether the analyses reflects a majority/minority perspective/range from the positive/negative sentiment analyses, and if there are changes in the articulated positions over time - so some trend analyses
- 2.To get a sense of the section of the population that is expressing these views – representative of region, age/generation, gender, location, profession etc?
- 3.To see the regional characteristics (Asia, Latin America, Caribbean, Africa, Middle East, East Asia, Pacific, Central Asia, Eastern Europe etc) for topics to resonate – e.g. freedom of movement, reproductive rights, political participation, economic autonomy etc.

Starting with the World Values Survey questions used for the Global Social Norms Index (GSNI):





Dimension	Indicator	Choices	Defining bias
Politics	<i>Men make better political leaders than women do</i>	Strongly agree, agree, disagree, strongly disagree	Strongly agree and agree
	<i>Women have the same rights as men</i>	1 not essential to 10 essentials	Intermediate form: 1 to 7
Education	<i>University is more important for a man than for a woman</i>	Strongly agree, agree, disagree, strongly disagree	Strongly agree and agree
Economic matters	<i>Men should have more right to a job than women</i>	Strongly agree, agree, disagree, strongly disagree	Strongly agree and agree
	<i>Men make better business executives than women do</i>	Agree, neither disagree	Agree
Physical integrity	<i>Is it ever justifiable: For a man to beat his wife</i>	1 never to 10 always	Stronger form: 2 to 10
	<i>Is it ever justifiable: Abortion</i>	1 never to 10 always	Weakest form: 1

The above four dimensions would also be tracked with the GSNI, so it would be interesting to see if there is resonance between the online discourse and the WVS trends in the GSNI. In addition, it would be good to track popular sentiment on issues that impact gender equality and women's rights: e.g. negative and positive sentiments around gender roles in the workplace, in politics, in households, in education, STEM and career aspirations, care work etc. Some of these are listed below:

- oWomen in workplace – issues around gender in employment e.g. sector segregation, concentration and glass ceiling
- oWomen in politics – parity, power and voice – issues around nominal representation to substantive participation via quotas, affirmative action, norm shifts etc
- oWomen's higher education and STEM career options – compatibility with familial roles, marriage and childbearing
- oWomen's responsibility for family stability and demographic transitions – responsibility for childbearing and childrearing
- oCare responsibility and women's work
- oNon-binary roles and rights – rejection of stereotypical heteronormative expectations and rights protection
- oHate speech and gender-based violence – misogynistic representation and GBV
- oWomen in public spaces – physical safety, freedom of movement and responsibility for GBV

Update as of June 02, 2022

I. Pilot Analysis Scope: Social Media Scanning (Exploratory Analysis)

- Platform: Twitter





- Countries: Colombia, Philippines, Uganda (pilot countries to be expanded with feedback from Regional Bureaus). Countries have been selected on the basis of external and internal factors:
 - o To account for regional diversity
 - o To test social media response on specific country current events:
 - Colombia: Legalization of abortion in February 2022
 - Philippines: Presidential elections in May 2022 featuring a prominent female candidate
 - o To test combining existing social media analysis and learning from previous studies: Uganda: UN Global Pulse and UNDP Uganda study

Country	No. of Twitter Users	Rationale for Country Selection
Colombia	4.3M (9% population)	Current events: Legalization of abortion in February 2022
Philippines	10.5M (9% population)	Current events: Presidential elections in May 2022 featuring prominent female candidate
Uganda	0.5M (9% population)	Previous studies by UN Global Pulse and UNDP Uganda on women and social media using radio data (2017)

- Languages: English, Spanish, Filipino
- Time period: Real-time tracking + retrospective data analysis (1 year historical data)
- Topic focus: Hate speech directed at women & girls

· Data Collection:

o Tweets from the pilot countries will be identified based on geocoding tools

Data will be collected using keyword queries based on a list of terms developed by the UNDP Gender Team and existing English, Spanish and Filipino language hate speech data sets from <https://github.com/leondz/hatespeechdata> ; after the initial search,

o additional co-occurring terms found in the process will be added in order to improve coverage of Twitter data

Characterizing Hate Speech: What is said and why?

o A hate speech classifier (algorithm) will be implemented to classify tweets between hate speech and non-hate speech

o Tweets detected with hate speech will be classified into several categories/themes

o Topic modelling will be used to identify dominant themes

- Ideally hate speech tweets could be further classified into the following sub-topics (if sufficient data is available):





- Women/girls + education
- Women/girls + work
- Women/girls + political participation · Women/girls + reproductive rights
- o A gender classifier (algorithm) will also be implemented
- To explore: Combining with radio data, c/o UN Global Pulse radio data monitoring

II. Timeline

April-May 2022	<ul style="list-style-type: none"> · Mapping of initiatives & literature review, development of project plan · Engagement across UNDP/UN & external entities: Country offices, SDGI, Crisis Bureau, UN Global Pulse, etc · Data testing, exploratory data analysis
June 2022	<ul style="list-style-type: none"> · Development of AI EWS prototype · Assess feasibility of lexicon, build data architecture, build models/algorithms for hate speech detection & topic modelling · Engagement with country offices for feedback · Build dashboard/ monitoring tool
July 2022	<ul style="list-style-type: none"> · (Early July) User testing of AI EWS & Dashboard: presentation and feedback · Integration to the Global Observatory platform for internal usage (with UNDP staff access rights only) · (End July) Go live of tool for selected audience
August 2022	<ul style="list-style-type: none"> · AI EWS ongoing monitoring & maintenance

III. Outputs

- Hate Speech Monitor oA dashboard will be developed to visualize the results, to be used for ongoing monitoring, featuring the following elements:

§ Real-time monitoring

§ Historical analysis: 1 year (June 2021- June 2022)

The dashboard will serve as an internal UNDP monitoring tool only, used as an early warning system to alert policymakers on gender backlashes online

oQuantitative Statistics:

§ No. of tweets / country / category

§ User demographics / country / category (data availability will vary)

- Gender, location, age (TBD)

§ Date, time, frequency, distribution of tweets

oTopic Modelling & Analysis:

§ Identification of key themes/issues per category

§ Identification of emerging trends, key topics of conversation





oSentiment Analysis to quantify sentiments and attitudes according to the following:

§Positive: positive attitude towards the topic

§Negative: negative attitude towards the topic

§Neutral: neither positive nor negative attitude towards the topic

§Unknown: partial or unclear statements related to the topic

§NA: not applicable to the topic

•Topics & Key Words for Exploratory Analysis:

oWomen & education

§Woman or women, girl or girls, female or females

§Educate or educating or education or school or schooling or study or studies or studying or university or high school

oWomen & STEM education

§Woman or women, girl or girls, female or females

§Educate or educating or education or school or schooling or study or studies or studying or university or high school

§Science or technology or engineering or mathematics

oWomen & gender-based violence

§Woman or women, girl or girls, female or female or girlfriends or girlfriends or wife or wives

§Rape or sexual assault or sexual violence or sexual abuse or force sex or child marriage or children marriage or underage marriage or forced marriage or sex trafficking or child trafficking or children trafficking or female genital mutilation or female genital cutting or forced prostitution

Abuse or abused or abusing or attack or attacked or attacking or assault or assaulted or assaulting or beat or beating or beaten or drag or dragged or dragging or harass or harassed or grope or groped or groping or harassing or kick or kicked or kicking or punch or punched or punching or rape or

§raping or raped or slap or slapped or slapping or stalk or stalked or stalking or trafficking or trafficked or violence

oWomen & Reproductive Rights

§Woman or women, girl or girls, female or females

§Abortion or contraception or birth control or pill or IUD or unwanted pregnancy

oWomen & work (This category may have too many combined topics and may need to be tweaked for the analysis)

§Woman or women, girl or girls, female or females or mother or mothers or wife or wives §Work or working or career or job or employment or office or employ or employed or employment

§Ambition or success or failure or promotion or promoted or demotion or demoted § Salary or raise or pay





§ Child or children or kid or kids or toddler or toddler or baby or babies or infant or infants § Family or home or domestic

o Women & political participation

§ Woman or women, girl or girls, female or females

§ Lead or leader or leaders or leadership

§ Power or powerful

§ Politics

§ Government or governing or govern

· To explore: Combining with radio data, c/o UN Global Pulse radio data monitoring

· References:

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6864473/pdf/sensors-19-04654.pdf>

<https://arxiv.org/pdf/1804.05704.pdf> <https://www.turing.ac.uk/research/research-programmes/public-policy/online-hateresearch-hub>

<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0243300>

<https://hatespeechdata.com/>

