The Power of Peace Speech

Peace Speech Analysis

Dec 10, 2021

Team Members
Hongling Liu, Haoyue Qi, Xuanhao Wu, Yuxin Zhou, Wenjie Zhu

Advisor Group
Peter Thomas Coleman Professor, Psychology and Education
Allegra Chen-carral Program Manager, Sustaining Peace Project
Larry Liebovitch Professor, Psychology and Physics
Philippe Loustaunau Managing Partner, Vista Consulting LLC

Project Description

Img Source: https://peacenews.info/sites/default/files/2021-09/old_PNs_large.jpg

Peacekeepers are actively conducting research on worldwise hate speech including **news articles and blog posts** in seeking to maintain robust and peaceful communities. But the peace speeches were commonly neglected in past researches.

Capstone students in the previous years had worked with Professor Coleman's team to fill in this missing side of the story by **finding connections between speech and the peacefulness of a country**.

This year, we will continue this research done by previous year's capstone students with a much richer dataset and **two new initial objectives**.

Project Description - *Goals*

Img Source: https://peacenews.info/sites/default/files/2021-09/old_PNs_large.jpg

Objective 1 - Contextual Task

- Explore the performance of different models on classifying high-peace and low-peace countries' English article embeddings encoded by BERT.

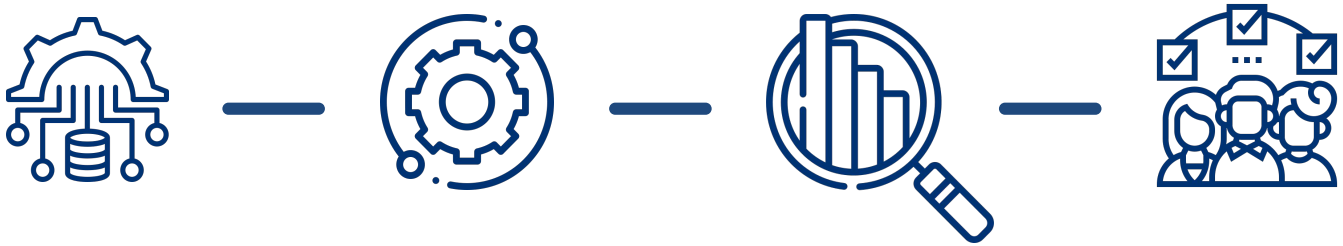
Premise 1: If there exists a high performance model, then there are language differences in the articles from countries with different peace levels.

Objective 2 - Contextless Task

- Study how the words' order affect the previous classification results by evaluating the model and encoder's performance over randomly shuffled articles.

Premise 2: If we see a performance drop after shuffling, then the sentence context is a strong peacefulness indicator.

Outline



Data EDA & Cleaning

Data Description
Data Engineering
Exploratory Data Analysis
Preprocessing

Experiment Setup

Train & Validation Split
Model Selection & Structure

Results & Analysis

Results Comparison
Problem Elimination
Refine Analysis
Visualize Learning Results
Side Notes

Conclusion

Key Findings Recap
Future Works

Data EDA & Cleaning - *Data Description*

Data Provider LexisNexis

Storage Location AWS S3 bucket

Storage Format

- Articles are stored as .json files under the *lowPeace* and *highPeace* directories.
- Each directory contains articles from 10 extreme countries by the *rule-of-5 metric*[1].
- Each .json object is one data point.

lowPeace		
Afghanistan	Iran	Uganda
Congo	Kenya	Zimbabwe
Guinea	Nigeria	
India	Sri Lanka	

highPeace		
Austria	Denmark	Norway
Australia	Finland	Sweden
Belgium	Netherlands	
Czech Republic	New Zealand	

Data Size ~33M articles from high peace countries and ~57M articles from low peace countries.

Columns of Interest title, content, wordCount, country

Computation Platform AWS SageMaker

1. For each country, scores for indices such as Global Peace Index, the Positive Peace Index, etc., are computed. Then countries are placed into low-, mid-, and high-peace categories for each index. If a country has at least 5 indices categorized as low-peace, and no index in high-peace, then the country is low-peace. Same logic follows for high-peace.

Data EDA & Cleaning - *Data Engineering*

Problem Identification - *The Small File Problem*

We have millions of files with 20 KB per file on average. The milliseconds to open, read, and close each file could accumulate to a large runtime overhead if a program has to execute those steps in high frequency. This **adds difficulties for us to view the dataset entirely and perform further analysis and operations**.

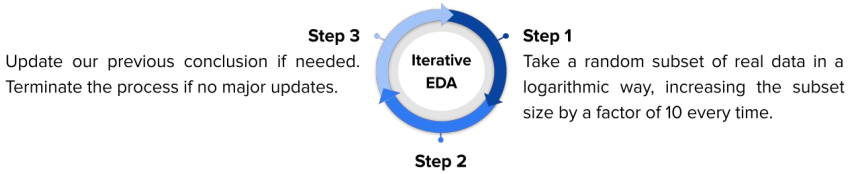
Solution - *Restructuring and Runtime Optimization*

1. Read in a large set of files from the real dataset at once and save them together as one large file for re-use in the rest of this project.
2. Use **UltraJSON** (ujson) package for faster content loading.
3. Parallelize the I/O bound loading task via **multithreading**.
4. Switch to a larger notebook instance (**ml.m5.4xlarge**) with a wider network bandwidth.

Optimization Results

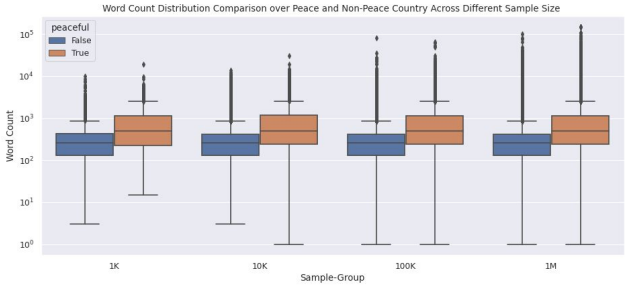
1. ~2.5 seconds to load 1K data points for each peace category, which is a **24x** speed up overall.
2. Downsampled 1M data from each peace category and stored them as one large .json file.

Data EDA & Cleaning - *Exploratory Data Analysis*



country	sample_perc	avg_wordCount	wordCount_CI_95
Afghanistan	0.2328	221.809708	(213.6184, 230.001)
Congo	0.4985	489.747041	(477.5319, 501.9622)
Guinea	0.0001	3121.000000	(nan, nan)
India	91.7565	422.663411	(421.0335, 424.2933)
Iran	2.3680	484.964443	(479.5853, 490.3436)
Kenya	0.5157	607.522397	(594.3597, 620.6851)
Nigeria	2.7022	579.226482	(571.1824, 587.2705)
Sri Lanka	1.3659	831.227542	(810.2969, 852.1582)
Uganda	0.2902	575.023777	(560.7757, 589.2719)
Zimbabwe	0.2701	566.279156	(551.8202, 580.7381)

country	sample_perc	avg_wordCount	wordCount_CI_95
Australia	76.0521	884.916874	(882.2838, 887.55)
Austria	0.4210	1240.742993	(1200.2169, 1281.2691)
Belgium	5.3665	703.747545	(683.1177, 724.3774)
Czech Republic	0.5935	415.039596	(402.2787, 427.8005)
Denmark	1.3845	1281.044854	(1252.5792, 1309.5105)
Finland	1.0927	1807.282694	(1767.3396, 1847.2258)
Netherlands	0.8204	1566.497075	(1530.0766, 1602.9176)
New Zealand	9.0610	526.680709	(522.3605, 531.0009)
Norway	1.2701	1851.772459	(1815.7198, 1887.8251)
Sweden	3.9382	1688.450307	(1673.9758, 1702.9249)



Insight 1

Heavy data imbalance between countries in each class.
Data Majorities: Australia & India
Data Minorities: Other countries

➜ Balance by country is less feasible. **Balance by class** strategy would be our top choice for most of the cases.

Insight 2

Significant length difference between articles from peace countries vs. non-peace countries.

➜ Small max length (128 tokens) to confound the length difference between +/- samples.

Data EDA & Cleaning - *Preprocessing*

Goal

Minimal preprocessing to **avoid destructions** to natural language structure. Stopwords are kept to **preserve negations and sentence sentiments**.

Cleaning Procedure

1. URL and Email Pattern identification and Removal
2. Unifying English versions via Dictionary Lookup
3. Named Entity Removal via SpaCy
4. BERT's Paired Preprocessor and Tokenization

Shuffling Procedure

1. Split cleaned sentence by space.
2. Randomly shuffle the sentence.

Current Challenge

1. SpaCy package is **unable to identify capitalize country names** and **country adjectives**, which would cause the cleaned sentences to still carry some **named entity residuals**.

Before preprocessing

Sadia Dada, Director Communications PMPKL, said, "The challenge of littering is not new to our country nor the efforts to combat it. Lack of awareness and infrastructure for disposal are key drivers abetting this bad habit adding that most important is how minor changes in the way we work can create room for each and every individual of society to be a part of it."\n\nPublished by HT Digital Content Services with permission from Daily Times. For any query with respect to this article or any other content requirement, please contact Editor at contentservices@htlive.com

After cleaning

, Director, said," The challenge of littering is not new to our country nor the efforts to combat it. Lack of awareness and infrastructure for disposal are key drivers abetting this bad habit adding that most important is how minor changes in the way we work can create room for each and every individual of society to be part of it." Published by with permission from.

After shuffling

of important be each of key in to country abetting this it." of that way Director, from. society combat said," and room challenge minor every littering it. bad habit awareness nor part are of for we , how our not drivers disposal the efforts most create adding infrastructure the can for is The Lack and to individual Published permission to changes with by is work new

Experiment Setup - *Train & Validation Split*

Approach 1 - *Random Split*

- Classical approach to train and validate a model
- Take same amount of samples from each class and ignore the data balance between countries within each class.

Approach 2 - *Split-by-Country*

- Train on samples from Australia & India, validate on samples from the rest of countries.
- **Alleviate the effect of named entity residuals** in the model's decision function.

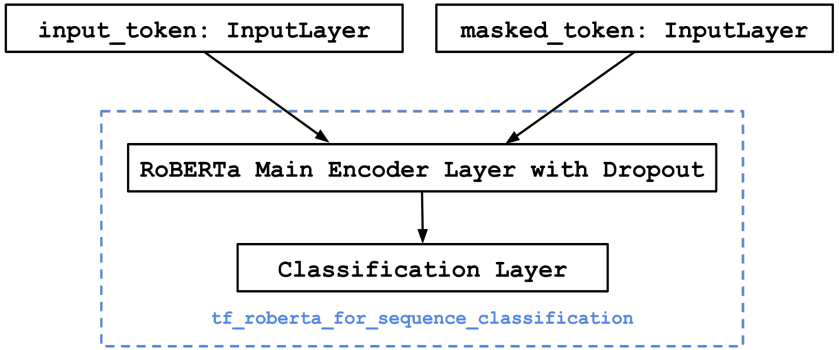
Premise 1: If there exists a **high performance model**, then there are **language differences** in the articles from countries with different peace levels.

Further Presumption: If all countries in the same peace group **sharing the same language structures**, then model's **performance would not weaken** by changing train & test split approaches.

Experiment Setup - *Model Selection & Structure*

Encoder Selection Criterion

- Pretrained on news articles.
- High ability to interpret sentiments.
- Reliable model provider and maintainer.



Model Structure for Fine-Tuning

- Add a classification head after the encoder layer.
- Same as passing the encoded sentence embeddings to a **logistic regressor**.

Model	Pro	Con
Logistic Regression	• Easy to implement (Model fine-tuning structure)	• Performance depends on whether data is linearly separable
XGBoost	• High performance • Make no assumptions of the data	• Can overfit data • Requires more computational resources • Interpretability depending on input features
SVM	• Effective in high dimensional spaces.	• Might not converge on large dataset • Might be difficult to find a good kernel function

Results & Analysis - *Results Comparison*

* Detailed stats in Appendix 1

Possible driver(s) of the performance drop:

1) Country imbalance under a random split method drive the model to focus on language structures from the majorities instead of all countries.

2) Countries are peace/non-peace in different ways.

Fine-tune RoBERTa on 8K samples and validated on 2K samples with 10 stratified folds.

Observations

1. Classification accuracy for all tests are ~70% and above.
   ➔ Model learned some weak language features.

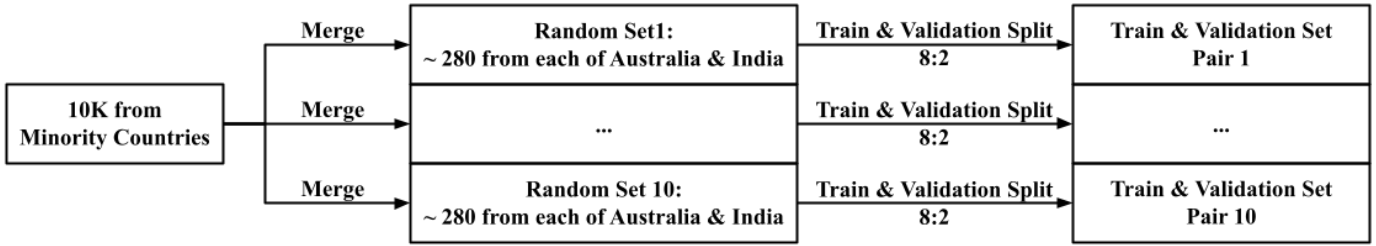
2. Performance on the original is better (~5%) than on the shuffled articles under same train-test split approach.
   ➔ Word context is not as important as words' individual meaning.

3. Performance on random split is significantly better than on split-by-country.
   ➔ Countries in the same peace group might has different language features.

Results & Analysis - *Problem Elimination*

Country-Balanced Random Split

1. Take a fixed number of articles from the minorities.
2. Measure the **median** of articles from each country in the minorities.
3. Add into this dataset the median number of articles from each of Australia and India.
4. Perform a random train-validation split over this compounded dataset.
5. Measure model's performance over 10 stratified trials.

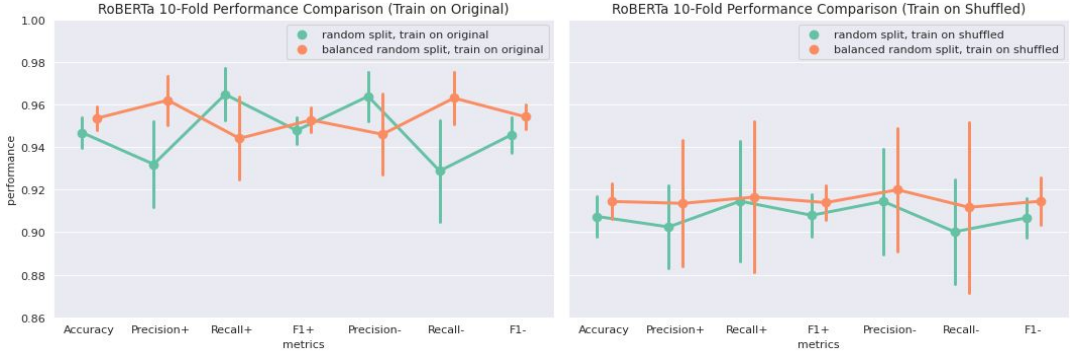


Implication

High performance by using this approach ➔ Country imbalance is not the cause of dropping in performance.

Then this would leave us the second driver to be the only cause for the performance drop for now.

Results & Analysis - *Refine Analysis*



* Detailed stats in Appendix 1

Observations

1. Performance before ≃ Performance after balancing the country for un/shuffled when splitting randomly.
   ➜ Country imbalance is not the cause of performance dropping.

2. Performance on the original is again better (~5%) than on the shuffled articles.
   ➜ Previous conclusion on the importance of word ordering is validated again.

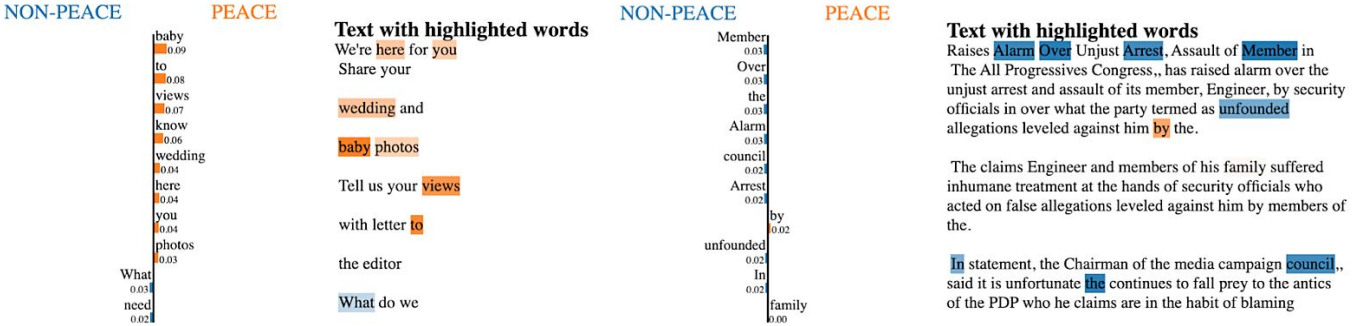
Conclusion

- **Countries are peace/non-peace in different ways** is the only possible cause for the performance drop when changing the split method we can conclude at this stage.
- Recall we saw F1- drop more than F1+ when split by country independently from whether shuffled or not. **There could be more structural variations in non-peace languages.**

Results & Analysis - *Visualize Learning Results*

Local **I**nterpretable **M**odel-Agnostic **E**xplanation [7]

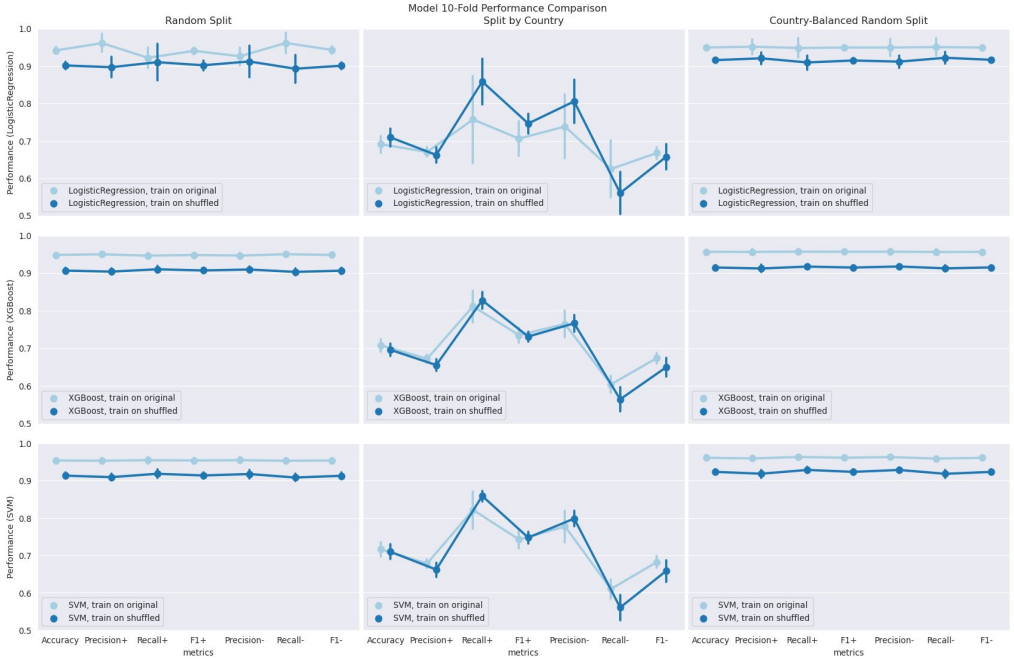
Pro	Con
• Make no assumption about the model to be explained. • State-of-arts technology for explaining deep learning models.	• Difficult to quantify feature importances globally. • Need human interpretation on the output. • Output is non-deterministic. • Slow to compute.



Results

- No obvious differences in results between experiment settings.
- Highlighted keywords agree with human judgement.

Results & Analysis - *Side Notes*



* Detailed stats in Appendix 2

Send the **[CLS]** token embedding to XGBoost or SVM to get final classification result.

Observations

1. Lines' y-axis position per column don't differ by a lot.
   ➜ Models' performance is "capped" by the performance at the fine-tuning stage.

2. Confidence interval at each point is narrower in the bottom two rows per column.
   ➜ XGBoost and SVM don't significantly improve the performance but add stability to it.

3. Validation Accuracy of SVM is slightly higher than that of XGBoost (*not obvious in the plot but is supported by actual stats in Appendix 2*).
   ➜ SVM is slightly more resistant to overfit.

Conclusion - *Key Findings Recap*

"If all countries in the same peace group sharing the same language structures, then model's performance would not weaken by changing train & test split approaches."
Linguistic features are a weak predictor of whether a country is high or low peace across countries. Countries may have different linguistic features relating to their level of peacefulness.

"If there exists a high performance model, then there are language differences in the articles from countries with different peace levels."
There still exists a weak link of language structures in articles from countries in the same peace group.

"If we see a performance drop after shuffling, then the sentence context is a strong peacefulness indicator."
Sentence context contributes a little but not significantly to the classification. Comparing to that, meaning of individual words might be more important.

Additional Comments:
RoBERTa is prone to overfit and sensitive to class imbalance. Country-wise balancing and complex classifiers can reduce the dispersion of prediction accuracy.

Conclusion - *Future Works*

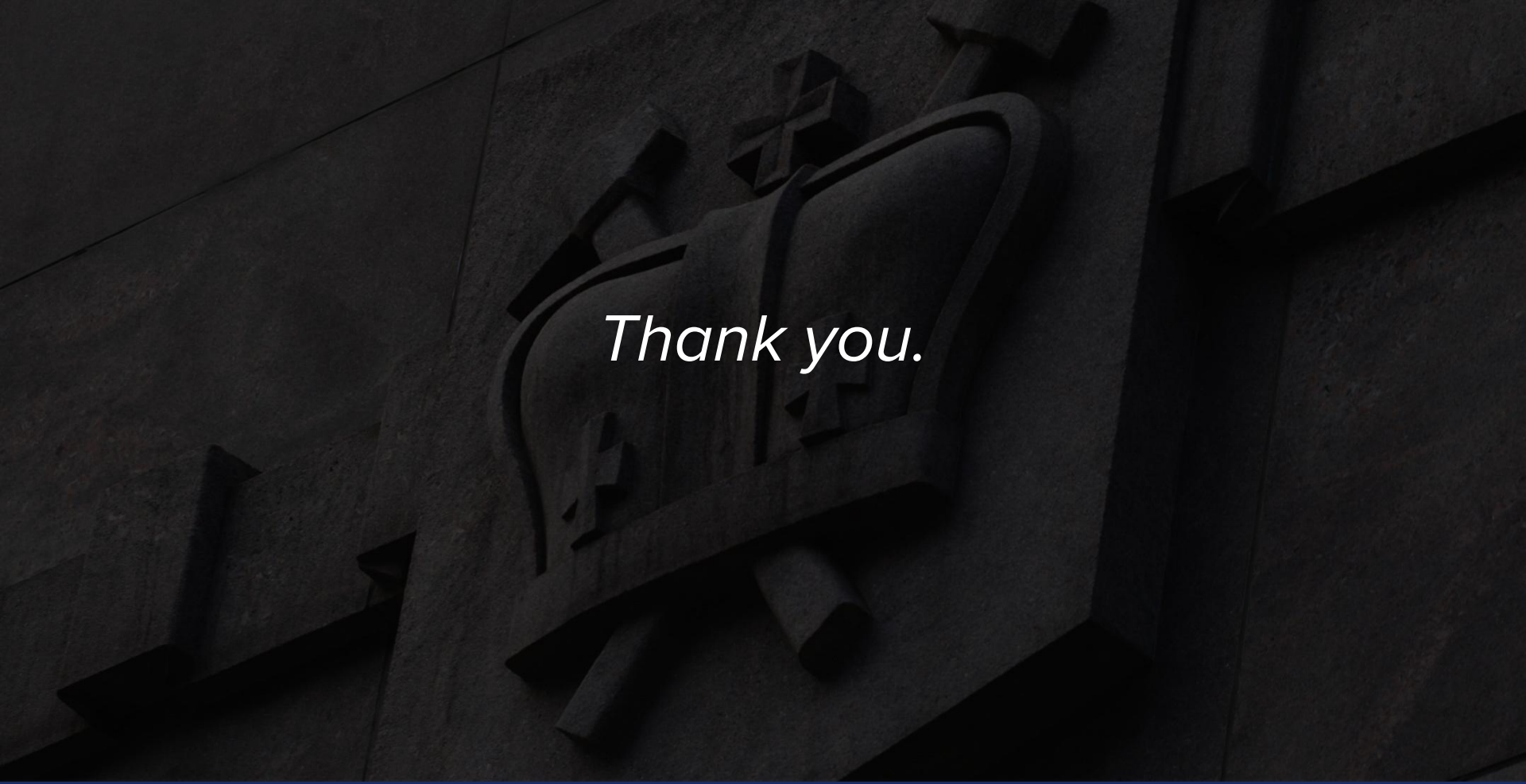
Img source: https://i.ytimg.com/vi/x2xYUTOcD-g/maxresdefault.jpg

Improve current pipeline - Train our own entity recognizer to better cleanup the sentence.

Explore other models - Try BERT encoders and other classification models to improve the accuracy.

Explain current model - Run current trained model on extreme countries identified by *matched-pair*[2] method and see if the results agrees with manual classification standard.

2. Cluster the countries using geography first, then select the highest and lowest peace countries in each geographical region, where the lowest peace country in a region is the one with the lowest number of indices classified as high-peace and highest number of indices classified as low-peace. Same logic follows for high-peace.


Thank you.

Reference

[1] Coleman, P. T., Fisher, J., Fry, D. P., Liebovitch, L. S., Chen-Carrel, A., & Souillac, G. (2020). How to live in peace? Mapping the science of sustaining peace: A progress report. The American psychologist, 10.1037/amp0000745. Advance online publication. https://doi.org/10.1037/amp0000745

[2] “Smart Batching Tutorial - Speed Up BERT Training · Chris McCormick,” Mccormickml.com, Jul. 29, 2020. https://mccormickml.com/2020/07/29/smart-batching-tutorial/.

[3] E. Alzahrani and L. Jololian, “How Different Text-preprocessing Techniques Using The BERT Model Affect The Gender Profiling of Authors,” arXiv.org, 2021. https://arxiv.org/abs/2109.13890.

[4] J. Camacho-Collados and M. T. Pilehvar, “On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis,” arXiv.org, 2017. https://arxiv.org/abs/1707.01780.

[5] Cathal Horan, “Tokenizers: How machines read,” FloydHub Blog, Jan. 28, 2020. https://blog.floydhub.com/tokenization-nlp/#wordpiece.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv.org, 2018. https://arxiv.org/abs/1810.04805.

[7] P. Goyal et al., “Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour,” arXiv.org, 2017. https://arxiv.org/abs/1706.02677.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why Should I Trust You?’: Explaining the Predictions of Any Classifier,” arXiv.org, 2016. https://arxiv.org/abs/1602.04938.

Appendix 1 - *Fine-Tune Statistics under different Train & Validation Split*

C.I and mean obtained from 10 Stratified Folds, 8K data in Train & 2K data in Validation per Fold.

		Accuracy	Precision +	Recall +	F1 +	Precision -	Recall -	F1 -
Original Sentence	Random Split	94.68 (94.13, 95.22)	93.19 (91.70, 94.69)	96.48 (95.56, 97.40)	94.78 (94.31, 95.25)	96.39 (95.53, 97.25)	92.88 (91.10, 94.65)	94.57 (93.94, 95.20)
	Split By Country	71.88 (70.17, 73.58)	67.64 (66.49, 68.79)	84.17 (78.04, 90.29)	74.77 (72.12, 77.43)	80.02 (75.59, 84.45)	59.59 (55.42, 63.76)	67.87 (66.04, 69.69)
	Country Balanced Random Split	95.35 (94.94, 95.77)	96.20 (95.34, 97.07)	94.40 (92.97, 95.85)	95.27 (94.85, 95.70)	94.60 (93.19, 96.01)	96.30 (95.40, 97.21)	95.42 (95.00, 95.84)
Shuffled Sentence	Random Split	90.74 (90.05, 91.43)	90.25 (88.80, 91.70)	91.46 (89.34, 93.58)	90.80 (90.06, 91.53)	91.45 (89.60, 93.30)	90.02 (88.19, 91.84)	90.67 (89.97, 91.37)
	Split By Country	68.72 (67.16, 70.28)	64.77 (63.41, 66.14)	82.47 (77.04, 87.90)	72.38 (70.35, 74.42)	76.94 (71.86, 82.02)	54.97 (50.43, 59.51)	63.57 (61.22, 65.93)
	Country Balanced Random Split	91.45 (90.83, 92.07)	91.36 (89.15, 93.57)	91.65 (89.02, 94.29)	91.39 (90.79, 92.00)	92.00 (89.86, 94.13)	91.17 (88.20, 94.14)	91.46 (90.63, 92.29)

Appendix 2 - *Model Performances Stats (Random Split)*

C.I and mean obtained from 10 Stratified Folds, 8K data in Train & 2K data in Validation per Fold.

		Accuracy	Precision +	Recall +	F1 +	Precision -	Recall -	F1 -
Original Sentence	Logistic Regression	94.19 (93.50, 94.87)	96.13 (94.34, 97.93)	92.20 (90.16, 94.24)	94.06 (93.36, 94.77)	92.61 (90.84, 94.38)	96.17 (94.13, 98.21)	94.29 (93.60, 94.98)
	XGBoost	94.86 (94.59, 95.13)	95.03 (94.66, 95.41)	94.67 (94.09, 95.25)	94.85 (94.57, 95.13)	94.70 (04.17, 95.23)	95.05 (94.65, 95.45)	94.87 (94.61, 95.14)
	SVM	95.35 (94.94, 95.77)	95.29 (94.80, 95.78)	95.43 (94.76, 96.10)	95.36 (94.94, 95.78)	95.43 (94.80, 96.06)	95.28 (94.77, 95.79)	95.35 (94.95, 95.76)
Shuffled Sentence	Logistic Regression	90.15 (89.38, 90.93)	89.68 (87.65, 91.72)	91.03 (87.41, 94.65)	90.20 (89.20, 91.20)	91.22 (88.13, 94.31)	89.28 (86.53, 92.03)	90.08 (89.42, 90.74)
	XGBoost	90.70 (90.12, 91.28)	90.42 (89.79, 91.05)	91.05 (90.40, 91.70)	90.73 (90.16, 91.31)	90..99 (90.35, 91.62)	90.35 (89.69, 91.01)	90.67 (90.08, 91.25)
	SVM	91.30 (90.67, 91.92)	90.90 (90.21, 91.58)	91.80 (90.92, 92.68)	91.34 (90.71, 91.98)	91.73 (90.89, 92.57)	90.80 (90.07, 91.53)	91.26 (90.63, 91.88)

Appendix 2 - *Model Performances Stats (Split-by-Country)*

C.I and mean obtained from 10 Stratified Folds, 8K data in Train & 2K data in Validation per Fold.

		Accuracy	Precision +	Recall +	F1 +	Precision -	Recall -	F1 -
Original Sentence	Logistic Regression	69.07 (67.42, 70.72)	67.01 (66.07, 67.96)	75.69 (67.00, 84.38)	70.58 (67.15, 74.02)	73.82 (67.53, 80.11)	62.45 (56.81, 68.10)	66.75 (65.58, 67.91)
	XGBoost	70.81 (69.56, 72.05)	67.22 (66.47, 67.98)	81.19 (78.07, 84.31)	73.51 (71.96, 75.06)	76.49 (73.89, 79.09)	60.42 (58.81, 62.03)	67.43 (66.37, 68.48)
	SVM	71.56 (70.16, 72.96)	67.79 (66.95, 68.64)	82.14 (78.44, 85.84)	74.22 (72.44, 76.00)	77.70 (74.53, 80.86)	60.98 (59.06, 62.91)	68.20 (67.05, 69.35)
Shuffled Sentence	Logistic Regression	70.93 (69.13, 72.72)	66.21 (64.65, 67.76)	85.83 (81.25, 90.41)	74.63 (72.68, 76.57)	80.54 (76.28, 84.80)	56.02 (51.86, 60.19)	65.71 (63.16, 68.27)
	XGBoost	69.61 (68.39, 70.83)	65.56 (64.40, 66.72)	82.78 (81.16, 84.40)	73.15 (72.15, 74.14)	76.67 (75.05, 78.29)	56.44 (54.05, 58.83)	64.96 (63.15, 66.76)
	SVM	70.94 (69.43, 72.45)	66.17 (64.76, 67.58)	85.86 (84.82, 86.90)	74.73 (73.60, 75.85)	79.81 (78.26, 81.37)	56.02 (53.46, 58.58)	65.80 (63.64, 67.95)

Appendix 2 - *Model Performances Stats (Country-Balanced Random Split)*

C.I and mean obtained from 10 Stratified Folds, 8K data in Train & 2K data in Validation per Fold.

		Accuracy	Precision +	Recall +	F1 +	Precision -	Recall -	F1 -
Original Sentence	Logistic Regression	94.95 (94.56, 95.34)	95.17 (93.67, 96.66)	94.83 (92.90, 96.75)	94.94 (94.52, 95.37)	94.94 (93.23, 96.66)	95.06 (93.34, 96.77)	94.95 (94.55, 95.35)
	XGBoost	95.69 (95.49, 95.89)	95.68 (95.15, 96.20)	95.74 (95.24, 96.23)	95.70 (95.53, 95.87)	95.72 (95.21, 96.23)	95.65 (95.09, 96.22)	95.68 (95.44, 95.92)
	SVM	96.08 (95.90, 96.26)	95.90 (95.55, 96.25)	96.29 (95.92, 96.66)	96.09 (95.93, 96.26)	96.26 (95.85, 96.67)	95.87 (95.53, 96.21)	96.07 (95.87, 96.26)
Shuffled Sentence	Logistic Regression	91.57 (91.14, 92.01)	92.07 (90.95, 93.18)	90.95 (89.53, 92.37)	91.48 (90.94, 92.01)	91.17 (89.95, 92.40)	92.20 (91.02, 93.39)	91.66 (91.29, 92.03)
	XGBoost	91.53 (91.10, 91.95)	91.26 (90.51, 92.00)	91.76 (91.45, 92.08)	91.51 (91.02, 92.00)	91.79 (91.55, 92.03)	91.30 (90.65, 91.94)	91.54 (91.15, 91.93)
	SVM	92.31 (91.81, 92.81)	91.81 (91.03, 92.60)	92.82 (92.19, 93.45)	92.31 (91.77, 92.85)	92.82 (92.31, 93.32)	91.79 (90.98, 92.59)	92.30 (91.82, 92.77)