Multilingual Joint Fine-tuning of Transformer models for identifying Trolling, Aggression and Cyberbullying at TRAC 2020

Sudhanshu Mishra, Indian Institute of Technology, Kanpur
Shivangi Prasad, University of Illinois Urbana Champaign
Shubhanshu Mishra, University of Illinois Urbana Champaign

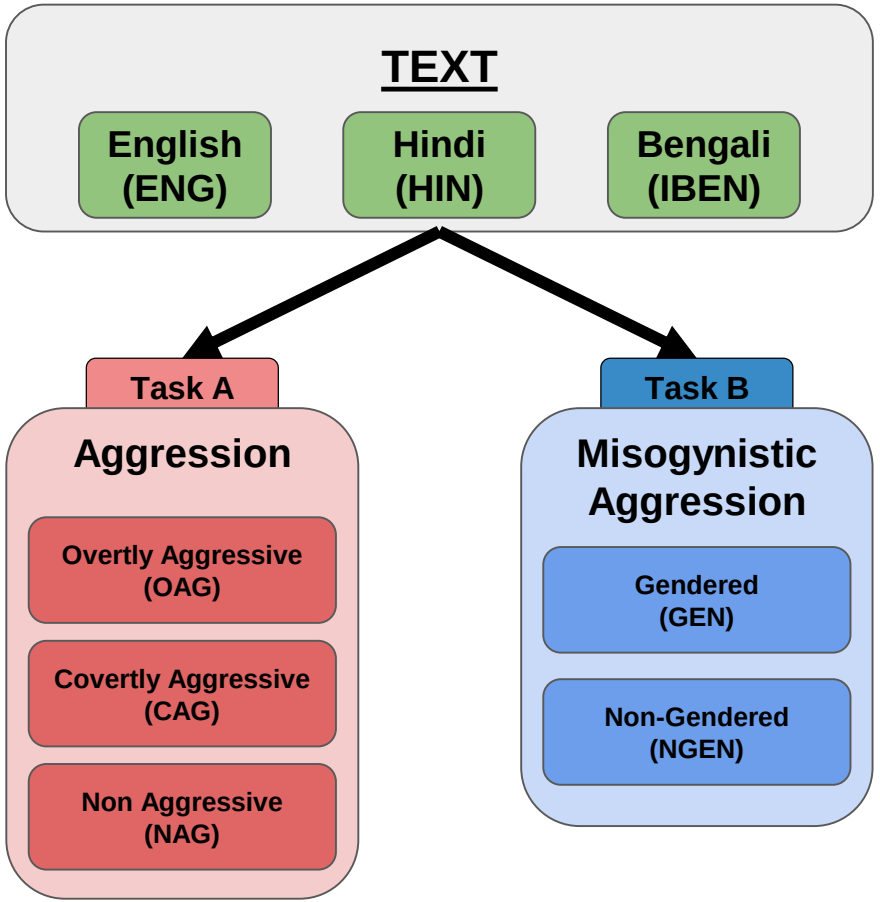
https://github.com/socialmediaie/TRAC2020

Our Contribution

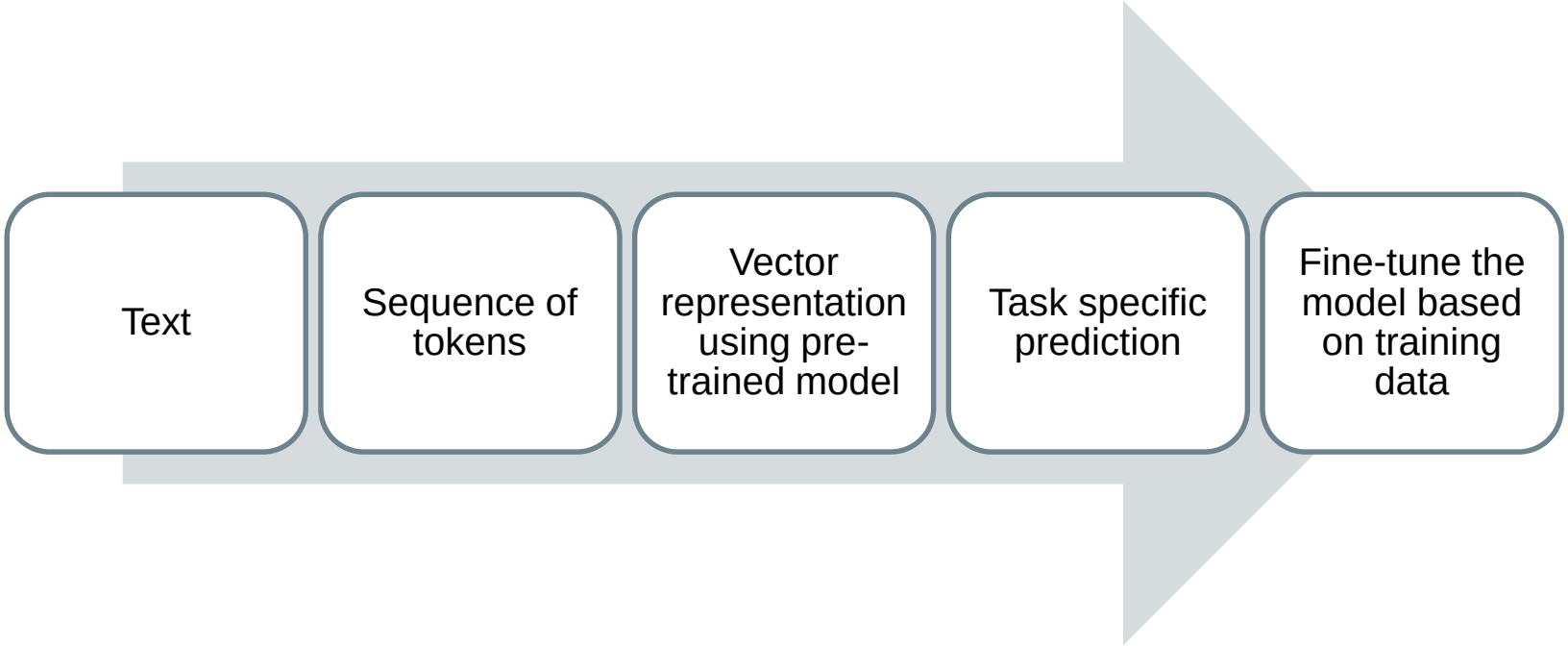
- Fine-tuned pre-trained transformer models on TRAC 2020 Aggression and Misogyny identification Task.
- Joint label training for solving both the tasks simultaneously. Inspired from our earlier work: https://github.com/socialmediaie/HASOC2019
- Multi-lingual models which predict on all languages.
- Open source code available at: https://github.com/socialmediaie/TRAC2020
- Pre-trained models + evaluation metrics available at: https://doi.org/10.13012/B2IDB-8882752_V1
- Models can be further fine-tuned and investigated. Details at GitHub page.

Task Description

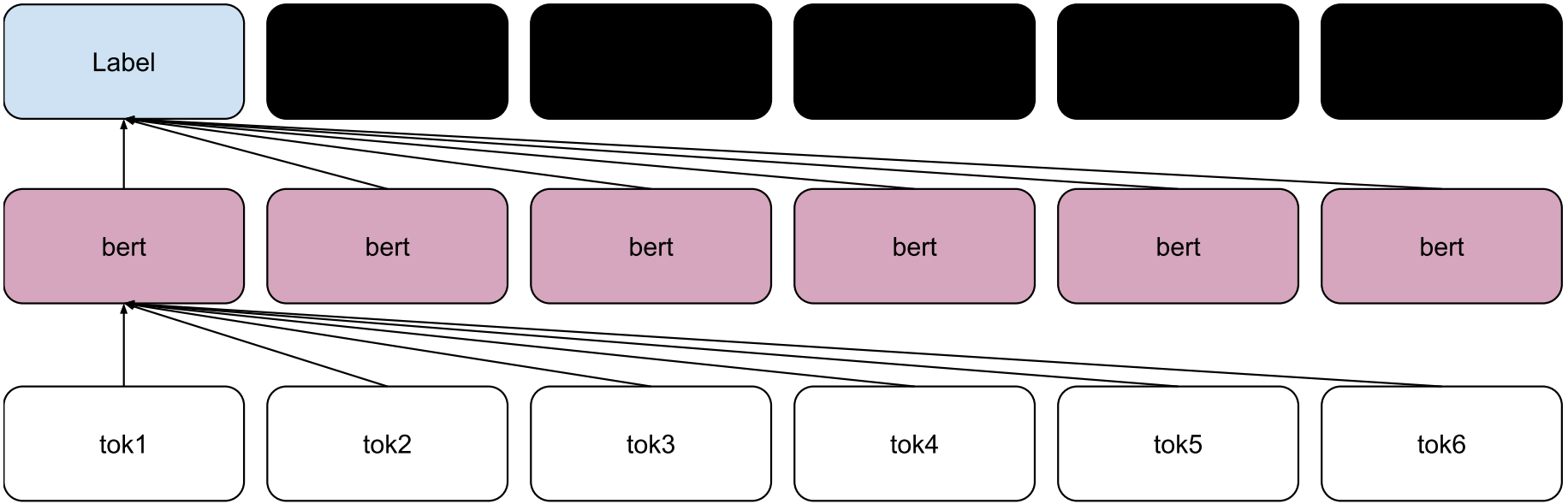
Lang	Task A			Task B		
	train	dev	test	train	dev	test
ENG	4263	1066	1200	4263	1066	1200
HIN	3984	997	1200	3984	997	1200
IBEN	3826	957	1188	3826	957	1188



Our Approach



Transformer models for classification



Fine-Tuning Techniques

For all of our experiments we fine tuned different transformer models using the **HuggingFace transformers** library. We tried three different methodologies:

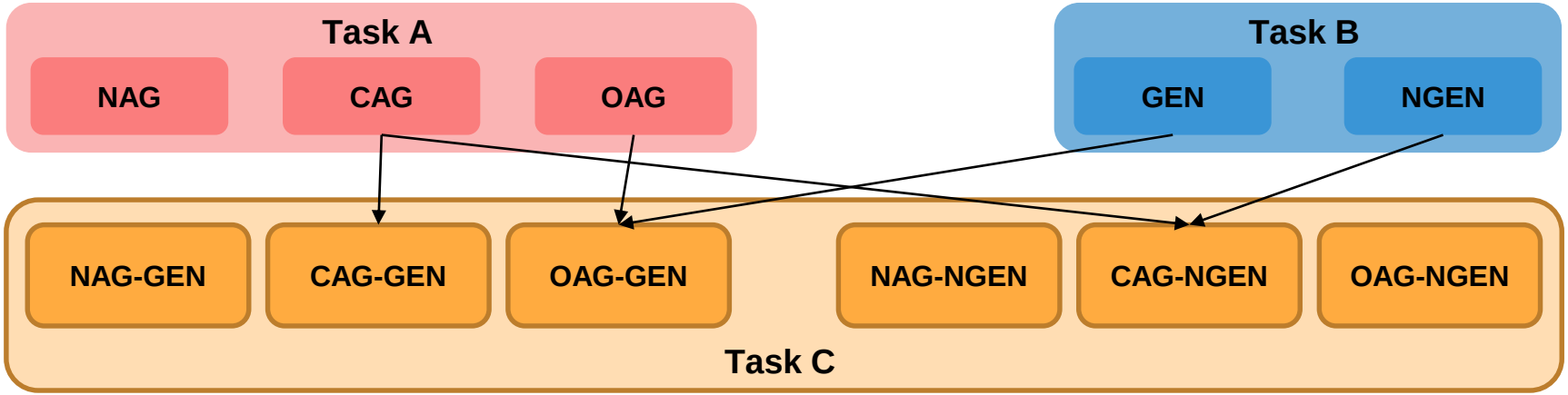
- Single task models
- Joint label training for both tasks **(C)**
- Marginalization of output probabilities **(M)**
- Joint training of all languages **(ALL)**

During our experiments, we tried different combinations of the above methods.

Top models based on dev set performance were submitted.

Joint Label Training (C)

If we combine the labels from all tasks, we can convert the original problem into six label classification problem, as shown below. (Similar to Mishra 2019) Model trained reduces the inference time computational cost to just doing inference with a single model.

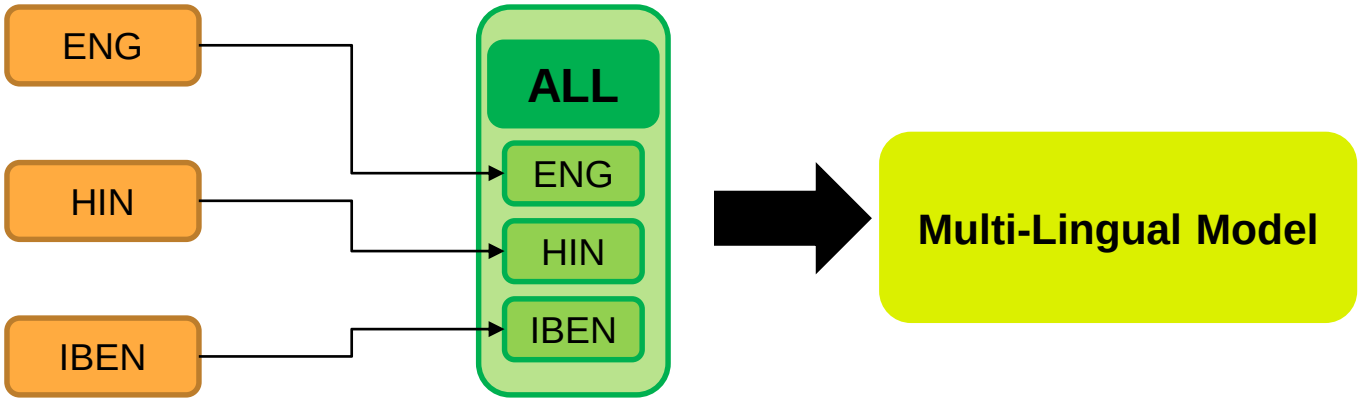


Marginalization of Labels (M)

The label probabilities from the joint label model (C) can be marginalized to get probabilities for each task label. *[Only used during inference, not training]*
This should result in a valid probability distribution for each task, which is useful for correct inference probability.

Joint training of all languages (ALL)

HIN and IBEN lack pre-trained models, but multi-lingual pre-trained models are available. We fine-tune those using training data from all languages.
This results in a single model for all languages, making the final model computationally cheaper to perform inference for all languages.



Experiments Conducted

We utilized various combinations of the techniques that we have mentioned. Following transformer models were fine-tuned.

ENG

- Bert-base-uncased
- Bert-base-cased
- Bert-base-multilingual-uncased
- Xlm-roberta-base

HIN

- Bert-base-multilingual-uncased
- Xlm-roberta-base

IBEN

- Bert-base-multilingual-uncased
- Xlm-roberta-base

A more detailed analysis of the experiments is mentioned in the systems paper.

Final Results

Our best performing model came:

- 2nd in 1/6 sub-tasks: ENG A
- 3rd in 3/6 sub-tasks: HIN A, B, and IBEN B.
- 4th in 1/6 sub-tasks: ENG B.
- The top models of ***Julian*** used ensemble based approach while all of our models required single model inference (sometimes even for all tasks and languages).
- The second best models of ***abaruah*** used SVM for top models.
- The closest approach to ours is ***Ms8qQxMbnjJMgYcw***, which trains BERT using multi-task setting, which performed similar to ours.

Final Results Task A

Lang	Model	Weighted F1		Rank		TRAC Rank
		dev	test	dev	test	
ENG	Bert-base-Multilingual-uncased (ALL)	0.798	0.728	1	3	-
	Bert-base-uncased (C)	0.795	0.759	2	2	-
	Bert-base-uncased (M)	**0.795**	**0.759**	**3**	**1**	**2**
	Overall Best Model	**-**	**0.802**	**-**	**-**	**1***
HIN	Bert-base-multilingual-uncased	0.708	0.778	1	3	-
	Bert-base-multilingual-uncased (ALL) (C)	**0.696**	**0.779**	**2**	**1**	**3**
	Bert-base-multilingual-uncased (ALL) (M)	0.695	0.778	3	2	-
	Overall Best Model	**-**	**0.812**	**-**	**-**	**1***
IBEN	**Bert-base-multilingual-uncased (ALL)**	**0.737**	**0.78**	**1**	**1**	**3**
	Xlm-roberta-base (M)	0.732	0.772	2	2	-
	Xlm-roberta-base (C)	0.731	0.772	3	3	-
	Overall Best Model	**-**	**0.821**	**-**	**-**	**1***

Final Result Task B

Lang	Model	Weighted F1		Rank		TRAC Rank
		dev	test	dev	test	
ENG	**Bert-base-uncased (M)**	**0.978**	**0.857**	**1**	**1**	**4**
	Xlm-roberta-base (ALL)	0.968	0.844	2	2	-
	Bert-base-multilingual-uncased (ALL) (M)	0.983	0.843	3	3	-
	Overall Best Model	**-**	**0.871**	**-**	**-**	**1***
HIN	Bert-base-multilingual-uncased	0.986	0.837	1	3	-
	Bert-base-multilingual-uncased (ALL)	**0.994**	**0.849**	**2**	**1**	**3**
	Bert-base-multilingual-uncased (ALL) (C)	0.962	0.843	3	2	-
	Overall Best Model	**-**	**0.878**	**-**	**-**	**1***
IBEN	**Bert-base-multilingual-uncased (ALL)**	**0.992**	**0.927**	**1**	**1**	**3**
	Bert-base-multilingual-uncased (ALL) (M)	0.965	0.926	2	2	-
	Bert-base-multilingual-uncased (ALL) (C)	0.902	0.925	3	3	-
	Overall Best Model	**-**	**0.938**	**-**	**-**	**1***

Re-use our **pre-trained models** for inference on your own data in just a few lines of code using **transformers** library

```
task, base_model = "ALL", "Sub-task C", "bert-base-multilingual-uncased"
base_model = f"socialmediaie/TRAC2020_{lang}_{lang.split()[-1]}_{base_model}"
tokenizer = AutoTokenizer.from_pretrained(base_model)
model = AutoModelForSequenceClassification.from_pretrained(base_model)
model.eval()

task_labels = ["OAG-GEN", "OAG-NGEN", "NAG-GEN", "NAG-NGEN", "CAG-GEN", "CAG-NGEN"]

sentence = """What a vacuum minded witch, product of May be so called Ranga-Billa. Such mean people gets Bookers Award, Disgusting!"""

processed_sentence = f"{tokenizer.cls_token} {sentence}"
tokens = tokenizer.tokenize(sentence)
indexed_tokens = tokenizer.convert_tokens_to_ids(tokens)

with torch.no_grad():
  logits, = model(torch.tensor([indexed_tokens]), labels=None)

preds_probs = softmax(logits.detach().cpu().numpy(), axis=1)
preds = np.argmax(preds_probs, axis=1)
preds_labels = np.array(task_labels)[preds]

print(f"Predicted: {preds_labels[0]}")
print(f"Probabilities: ")
dict(zip(task_labels, preds_probs[0]))
"""You should get an output as follows:

Predicted: OAG-GEN
Probabilities:
{'CAG-GEN': 0.011104056,
 'CAG-NGEN': 0.0018891948,
 'NAG-GEN': 0.013686359,
 'NAG-NGEN': 0.0017242465,
 'OAG-GEN': 0.9437853,
 'OAG-NGEN': 0.027810896}
"""
```

Conclusion

- We found fine-tuning pre-trained transformer models results in a competitive performance on all TRAC 2020 tasks.
- Joint label and multi-lingual training allowed us to address data sparsity issue and this was the best approach for HIN and IBEN datasets.
- ALL models are the computationally cheapest models for inference.
- Code for reproducing our results can be found at: https://github.com/socialmediaie/TRAC2020

References

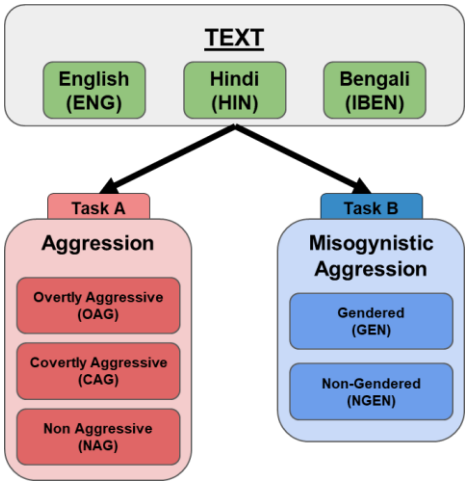
- Mishra, Sudhanshu, Shivangi Prasad, and Shubhanshu Mishra. 2020. “Multilingual Joint Fine-Tuning of Transformer Models for Identifying Trolling, Aggression and Cyberbullying at TRAC 2020.” In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying (TRAC-2020)*.
- Mishra, Shubhanshu, and Sudhanshu Mishra. 2019. “3Idiots at HASOC 2019: Fine-Tuning Transformer Neural Networks for Hate Speech Identification in Indo-European Languages.” In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, 208–13. Kolkata, India. http://ceur-ws.org/Vol-2517/T3-4.pdf.
- Mishra, Shubhanshu, Shivangi Prasad, and Shubhanshu Mishra. 2020. “Trained Models for Multilingual Joint Fine-Tuning of Transformer Models for Identifying Trolling, Aggression and Cyberbullying at TRAC 2020.” University of Illinois at Urbana-Champaign. https://doi.org/10.13012/B2IDB-8882752_V1.
- Mishra, Shubhanshu. 2019. “Multi-Dataset-Multi-Task Neural Sequence Tagging for Information Extraction from Tweets.” In *Proceedings of the 30th ACM Conference on Hypertext and Social Media - HT ’19*, 283–84. New York, New York, USA: ACM Press. https://doi.org/10.1145/3342220.3344929.

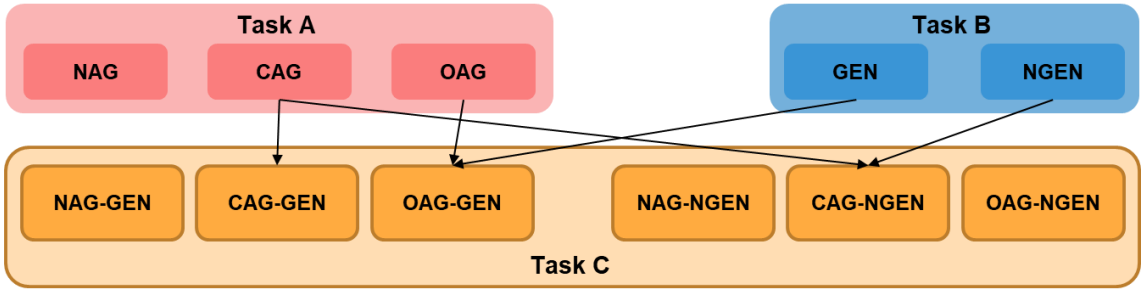
Thank You

- **Our paper:** Mishra, Sudhanshu, Shivangi Prasad, and Shubhanshu Mishra. 2020. “Multilingual Joint Fine-Tuning of Transformer Models for Identifying Trolling, Aggression and Cyberbullying at TRAC 2020.” In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying (TRAC-2020)*.
- More tools related to social media information extraction can be found at the SocialMediaIE project: https://socialmediaie.github.io/
- There will be a hands-on tutorial on Information extraction from social media using SocialMediaIE at LREC 2020: https://lrec2020.lrec-conf.org/en/workshops-and-tutorials/tutorials/
- Contact:
  - Sudhanshu Mishra: https://twitter.com/SudoMishra
  - Shivangi Prasad: https://twitter.com/shivangiPhy
  - Shubhanshu Mishra: https://shubhanshu.com/

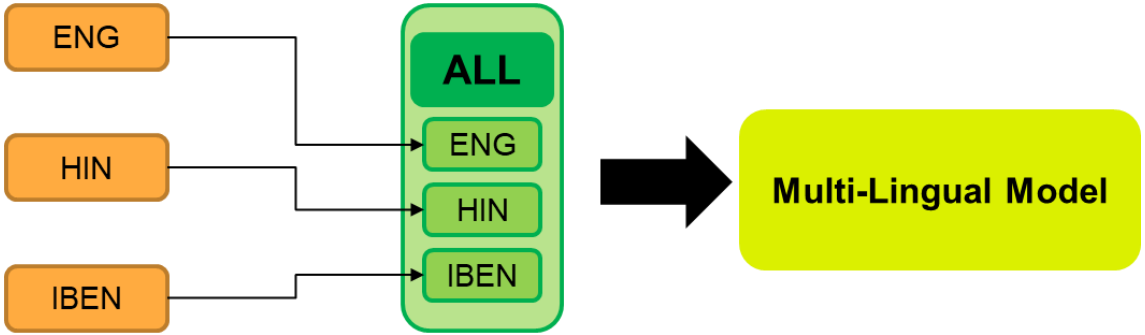
Multilingual Joint Fine-tuning of Transformer models for identifying Trolling, Aggression and Cyberbullying at TRAC 2020

https://github.com/socialmediaie/TRAC2020





$$P(\text{NAG}) = P(\text{NAG-GEN}) + P(\text{NAG-NGEN})$$



2nd in 1/6 sub-tasks: ENG A
3rd in 3/6 sub-tasks: HIN A, B, and IBEN B
4th in 1/6 sub-tasks: ENG B
Computationally faster and cheaper inference cost.