TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

TU WIEN Informatics

electronic commerce

Measuring explainability in hate speech detection using the HateXplain dataset

Seminar Talk, Markus Reichel 01529191
TUW NLP Seminar

Advisors: Gábor Recski, PhD
Univ.Prof. Dr. Allan Hanbury

Disclaimer

This presentation contains profane words.

Outline

- HateXplain, Rationales
- Explainability Theory
- ERASER benchmark
- Implementing ERASER
- First Results
- Further Work
- Discussion

HateXplain, Rationales

HateXplain

- Annotated Amazon MTurk dataset with 3-classes (hate, offensive, normal)
- Several target communities
- Annotation contains rationales

Rationales

„Human attention“

Show which part of the sentence is important for the decision

→ **Means of explainability**

Text	Dad should have told the muzrat whore to fuck off, and went in anyway
Label	Hate
Targets	Islam
Text	A nigress too dumb to fuck has a scant chance of understanding anything beyond the size of a dick
Label	Hate
Targets	Women, African
Text	Twitter is full of tween dikes who think they're superior because of "muh oppression." News flash: No one gives a shit.
Label	Offensive
Targets	Gay

MATHEW, Binny, et al. HateXplain: A benchmark dataset for explainable hate speech detection. arXiv preprint arXiv:2012.10289, 2020.

Used Explainability Framework

Local Post-Hoc	Explain a single prediction by performing additional operations (*after* the model has emitted a prediction)
Local Self-Explaining	Explain a single prediction using the model itself (calculated from information made available from the model *as part of* making the prediction)
Global Post-Hoc	Perform additional operations to explain the entire model's predictive reasoning
Global Self-Explaining	Use the predictive model itself to explain the entire model's predictive reasoning (*a.k.a.* directly interpretable model)

Table 1: Overview of the high-level categories of explanations (Section 3).

Modern deep learning architectures like BERT ad-hoc only locally self-explaining (trust?)

→ **Extract linguistic rules with a rule-based system**

Danilevsky, Marina, et al. "A survey of the state of explainable AI for natural language processing." arXiv preprint arXiv:2010.00711 (2020).

Used Explainability Framework

	Local Post-Hoc	Explain a single prediction by performing additional operations (*after* the model has emitted a prediction)
Rationales →	**Local Self-Explaining**	Explain a single prediction using the model itself (calculated from information made available from the model *as part of* making the prediction)
	Global Post-Hoc	Perform additional operations to explain the entire model's predictive reasoning
Rule Systems →	**Global Self-Explaining**	Use the predictive model itself to explain the entire model's predictive reasoning (*a.k.a.* directly interpretable model)

Table 1: Overview of the high-level categories of explanations (Section 3).

Modern deep learning architectures like BERT ad-hoc only locally self-explaining (trust?)

→ **Extract linguistic rules with a rule-based system**

Danilevsky, Marina, et al. "A survey of the state of explainable AI for natural language processing." arXiv preprint arXiv:2010.00711 (2020).

ERASER Framework

Evaluating **R**ationales **A**nd **S**imple **E**nglish **R**easoning

Young et al.

- Propose several metrics for predicted rationals

- Aim to capture two dimensions:
  1) How well rationales by models align with human rationales
  2) To which degree the rationales influence the prediction

- Provide an open source implementation on Github
- (Also provide example datasets & a leaderboard)

https://www.eraserbenchmark.com/

1) → „Plausibility“
2) → „Faithfulness“

DEYOUNG, Jay, et al. ERASER: A benchmark to evaluate rationalized NLP models. arXiv preprint arXiv:1911.03429, 2019.

Plausibility

Agreement with human rationales

Interpretation: How convincing the interpretation is to humans
Two variants: discrete and „soft“ selection

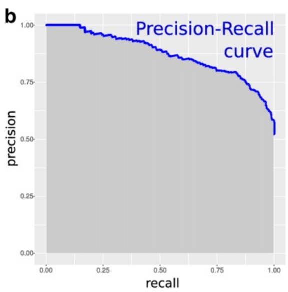
Discrete:

Intersection-Over-Union(IOU): for two spans,
Partial match = overlap/union > threshold [0.5]
IOU F1 = F1Score(all partial matches)
Token F1 = (token-level precision & recall)

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Continuous:

Area Under the Precision-Recall Curve (AUPRC)
Sweeping a threshold over token scores



DEYOUNG, Jay, et al. ERASER: A benchmark to evaluate rationalized NLP models. arXiv preprint arXiv:1911.03429, 2019.

Faithfulness

Influence of the rationales to the prediction

Interpretation: How accurately it reflects the true reasoning process of the model
Two metrics,

say **m(xi)** ist the probability that sentence xi is classified offensive
m(ri) is the probability that the predicted rationales ri alone are classified offensive
m(xi\ri) is the sentence with removed predicted rationales

Comprehensiveness:

(Were all features needed to make a prediction?)

- **= m(xi) – m(xi\ri)**
- The higher, the better (negative: model became more confident w/o rationales)

Sufficiency:

(Do extracted rationales contain enough signal?)

- **= m(xi) – m(ri)**
- The lower, the better

DEYOUNG, Jay, et al. ERASER: A benchmark to evaluate rationalized NLP models. arXiv preprint arXiv:1911.03429, 2019.

Faithfulness (cont.)

How to remove continous rationales?

→ Remove top k rationales (threshold)

- **Aggregation:**
- Motivated by saliency maps
- Group rationals in k=5 bins
- rik = rationale i up to and including bin k
- Top 1%, 5%, 10%; 20%, 50%
- „Area Over the Perturbation Curve“

$$\frac{1}{|\mathcal{B}|+1}\left(\sum_{k=0}^{|\mathcal{B}|} m(x_i)_j - m(x_i \backslash r_{ik})_j\right)$$

DEYOUNG, Jay, et al. ERASER: A benchmark to evaluate rationalized NLP models. arXiv preprint arXiv:1911.03429, 2019.

ERASER Output

```
Plausibility
IOU F1 : 0.1255215896343243
Token F1 : 0.4439984064957904
AUPRC : 0.588625850234053

Faithfulness
Comprehensiveness : 0.6083561550950038
Sufficiency 0.15281228368862493
```

If e.g. soft rationale is not in the input file (see later):

ERASER skips calculation

DEYOUNG, Jay, et al. ERASER: A benchmark to evaluate rationalized NLP models. arXiv preprint arXiv:1911.03429, 2019.

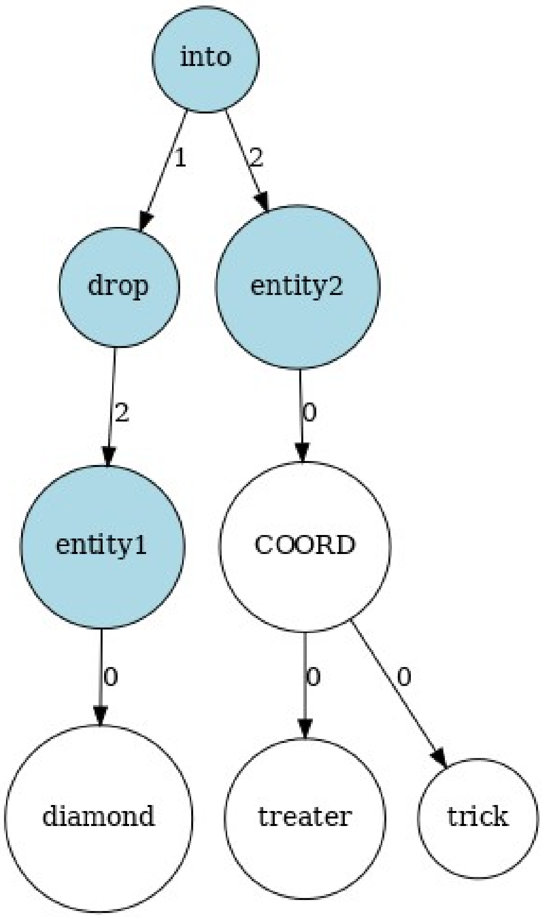
Applying the metrics to POTATO

Plausibility:

- Currently, hard predictions are implemented for IOU F1 & Token F1
- The predicted rationales are all words of matching rules → [„into“, „drop“, „entity1“, „entity2“]

Faithfulness:

- The probability function **m(x)** is between 0 and 1, deep learning logits are continous
- However, a potato rule matches either fully or not
- Single sentence faithfulness metrics are either 0 or 1 (Smoothed out by aggregation)

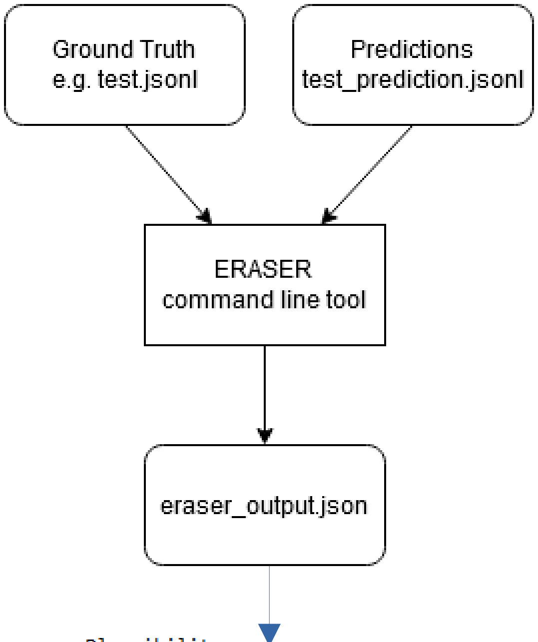


ERASER Input

Format

- jsonl
- Slightly different formats for ground truth and prediction
- Text is not in the jsonl but in the docs folder

```
—Model_Eval
 │    bestModel_bert_base_top5.json
 │    test.jsonl
 │    train.jsonl
 │    val.jsonl
 │
 └───docs
          1017919_gab
          1053393_gab
          1073224_gab
          1077853_gab
          1094959_gab
          1135780_gab
          1146615_gab
          1154520_gab
```




```
Plausibility
IOU F1 : 0.1255215896343243
Token F1 : 0.4439984064957904
AUPRC : 0.5886258502340532

Faithfulness
Comprehensiveness : 0.6083561550950038
Sufficiency 0.15281228368862493
```

ERASER Input (Ground Truth)


```json
{
  "annotation_id": "13851720_gab",
  "classification": "hatespeech",
  "evidences": [
    [
      {
        "docid": "13851720_gab",
        "end_sentence": -1,
        "end_token": 17,
        "start_sentence": -1,
        "start_token": 13,
        "text": "19424 11382 3489 2653"
      },
      {
        "docid": "13851720_gab",
        "end_sentence": -1,
        "end_token": 28,
        "start_sentence": -1,
        "start_token": 21,
        "text": "4654 3334 19269 1996 2175 10139 2213"
      }
    ]
  ],
  "query": "What is the class?",
  "query_type": null
}
```


https://www.eraserbenchmark.com/

ERASER Input (Prediction)


```
{
  "annotation_id": "13851720_gab",
  "classification": "hatespeech",
  "classification_scores": {
    "hatespeech": 0.9781582355499268,
    "normal": 0.003347641555924177,
    "offensive": 0.018494125455617905
  },
  "rationales": [
    {
      "docid": "13851720_gab",
      "hard_rationale_predictions": [
        {
          "end_token": 7,
          "start_token": 6
        },
        {
          "end_token": 37,
          "start_token": 36
        }
      ],
      "soft_rationale_predictions": [
        0.018977651372551918,
        0.018510917201638222,
        0.018933551385998726,
        0.4306974411010742
      ],
      "truth": 0
    }
  ],
  "sufficiency_classification_scores": {
    "hatespeech": 0.9711454510688782,
    "normal": 0.004742590710520744,
    "offensive": 0.024111928418278694
  },
  "comprehensiveness_classification_scores": {
    "hatespeech": 0.005441979970782995,
    "normal": 0.9660893678665161,
    "offensive": 0.028468627482652664
  }
}
```


= ~~m(xi)~~ m(ri)

= ~~m(xi)~~ m(xi\ri)

https://www.eraserbenchmark.com/

Calling ERASER

ERASER structure:

Just to important files:

rationale_benchmark/metrics.py Contains main() function

rationale_benchmark/util.py Contains documentation

Current way to call ERASER:

- Local copy in potato/scripts folder
- main() needs arguments
- Copied the content of the main function to runEvaluation
- Parameters are arguments

```
import rationale_benchmark.metrics as eraser

eraser.runEvaluation("None", # neutralclassname
                     data_dir=datadir, # data dir
                     split=testtrainorval, # split
                     results=pathtopredictions, # results
                     score_file=datadir+"/eraser_output.json", # score
                     strict=False) # strict
                     #iou_thresholds=[0.5], # iou
                     #aopc_thresholds=[0.01, 0.05, 0.1, 0.2, 0.5]) # aopc
```

In evaluation script:

```
print_classification_report(df, stats)
print("------------------------")
matched_result = evaluator.match_features(df, features[target])
subgraphs = matched_result["Matched rule"]
labels = matched_result["Predicted label"]
data_tsv_to_eraser(file)
prediction_to_eraser(file, subgraphs, labels, labels, labels, target)
call_eraser("./hatexplain", "val", "./hatexplain/val_prediction.jsonl")
print("------------------------")
```

Applying the metrics to HateXplain

```
"classification": "hatespeech",
"classification": "offensive",
"classification": "offensive",
"classification": "offensive",
"classification": "offensive",
"classification": "hatespeech",
classification": "hatespeech",
"classification": "hatespeech",
"classification": "hatespeech",
"classification": "hatespeech",
```

Rationales

- Only available for hatespeech/offensive classes
- HateXplain just discards all non-hate ground truth data

Dirty hack

Add normal label in metrics.py	2 years ago

- Hardcoded normal class in ERASER metrics.py

```
286  +      labels +=['normal']
```

Advantage of discarding:

- We can in theory now directly compare our results to the HateXplain models

First Results of Plausibility

Model	IOU F1	Token F1
(BERT HXPlain) (test.jsonl)	(0.126)	(0.444)
Rules: sexism val.tsv	0.279	0.165
Rules: homophobia secondary_val.tsv	0.090	0.047

→ sanity check:
HateXplain BERT ran on original hatespeech/offensive/normal task

→ Rules ran on <target>/None task

PS: regarding testing: val is shorter than train

Model [Token Method]	**Explainability**		
	Plausibility		
	IOU F1↑	Token F1↑	AUPRC↑
CNN-GRU [LIME]	0.167	0.385	0.648
BiRNN [LIME]	0.162	0.361	0.605
BiRNN-Attn [Attn]	0.167	0.369	0.643
BiRNN-Attn [LIME]	0.162	0.386	0.650
BiRNN-**HateXplain** [Attn]	**0.222**	**0.506**	**0.841**
BiRNN-**HateXplain** [LIME]	0.174	0.407	0.685
BERT [Attn]	0.130	0.497	0.778
BERT [LIME]	0.118	0.468	0.747
BERT-**HateXplain** [Attn]	0.120	0.411	0.626
BERT-**HateXplain** [LIME]	0.112	0.452	0.722

→ AUPRC would need continuous rationale prediction (possible if smoothed out)

→ scores will be better with multi-rule matching

→ only single word nodes are returned, no ‚|‘ yet

(see later)

MATHEW, Binny, et al. HateXplain: A benchmark dataset for explainable hate speech detection. arXiv preprint arXiv:2012.10289, 2020.

Further Work

Important:

1) Multi-rule matching
2) Predicted labels to calculate Faithfulness
3) Support ‚|‘ (see homophobia rules)

Further Experiments:

1) Rationale smoothing to get AUPRC
2) Faithfulness: Different ways of masking words (<UNK>, parsing, etc.)
3) Integration into Potato?
4) Evaluate human annotators
5) Look at normalized ERASER metrics (Carton et al.)
6) Reimplement ERASER metrics
7) Create rule system for another target
8) Extend „HASOC 100% dataset“ with rationales

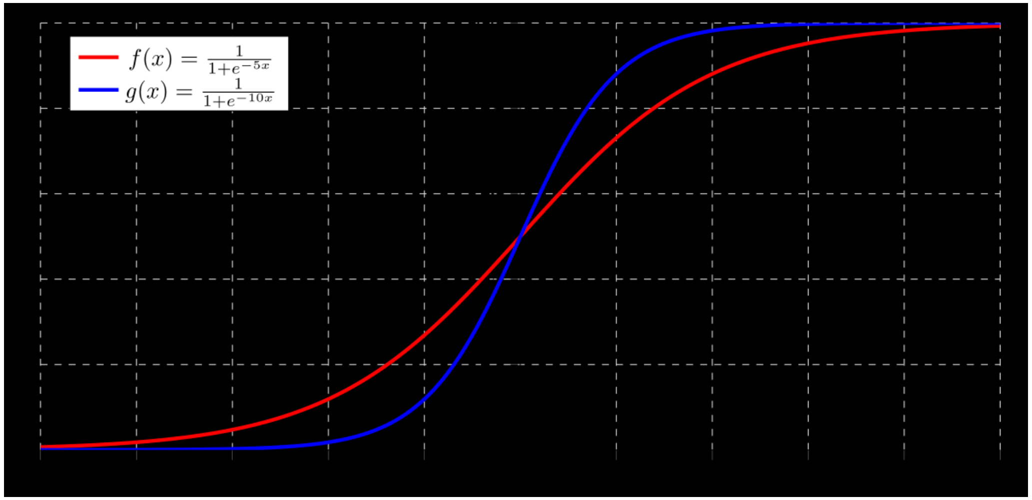
1) Rationale smoothing for AUPRC

Prediction:

[1, 0, 0, 0, 1, 0, 0,...]

→

[0.2324, 0.0111, 0.0024, 0.0032, 0.5342, …]



→ soft_prediction AUPRC score

2) Faithfulness Masking

m(xi\ri)

- **xi\ri = ???**

1) Swap rationales with e.g. <UNK> token and parse again
2) Mask nodes from existing graph
3) Remove from rationales sentence and parse again

3) Integration into Potato

- Currently, evaluate_HateXplain calls extern ERASER script
- Integration of evaluate_HateXplain.py: leave in scripts folder
- Create more general evaluate_rationals?

4) Evaluate human annotators

HateXplain already done by Carton et al.

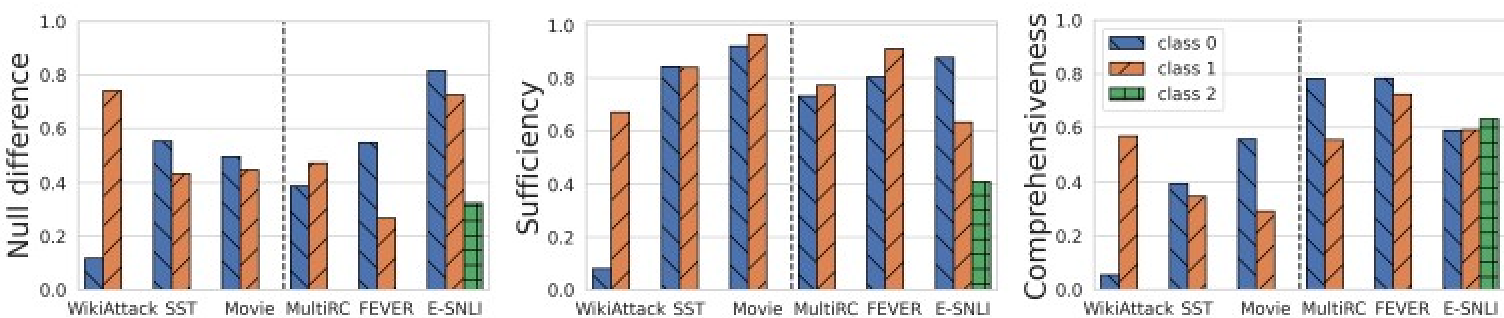
Maybe with a smaller dataset? → See 8)

CARTON, Samuel; RATHORE, Anirudh; TAN, Chenhao. Evaluating and characterizing human rationales. arXiv preprint arXiv:2010.04736, 2020.

5) Look at normalized ERASER metrics

Evaluate if needed, idea of a metric is to compare

Faithfulness	
Comp.↑	Suff.↓
0.316	**-0.082**
0.421	-0.051
0.278	0.001
0.308	-0.075
0.281	0.039
0.343	-0.075
0.447	0.057
0.436	0.008
0.424	0.160
0.500	0.004



(a) Null difference. (b) Normalized sufficiency. (c) Normalized comprehensiveness.

CARTON, Samuel; RATHORE, Anirudh; TAN, Chenhao. Evaluating and characterizing human rationales. arXiv preprint arXiv:2010.04736, 2020.

6) Reimplement ERASER metrics

→ No need for ERASER-specific input format
→ Room for extensions: new metrics, adapted metrics

7) Create rule system for another target

→ Are two rule systems enough?

There is another option...

8) … „100% dataset“ rationales

→ Was created by hand from HASOC data
→ Hate annotation is subjective
→ Inconsistent annotations were removed
→ 200 entries
→ Includes a rule system with 100% precision

Result of using all the rules: Precision: **1.000**, Recall: **0.855**, Fscore: **0.922**

Add rationales by hand too and run ERASER on it?

Questions / Discussion / Thank you!

Sources:

[1] DANILEVSKY, Marina, et al. A survey of the state of explainable AI for natural language processing. arXiv preprint arXiv:2010.00711, 2020.

[2] MATHEW, Binny, et al. HateXplain: A benchmark dataset for explainable hate speech detection. arXiv preprint arXiv:2012.10289, 2020.

[3] DEYOUNG, Jay, et al. ERASER: A benchmark to evaluate rationalized NLP models. arXiv preprint arXiv:1911.03429, 2019.

[4] CARTON, Samuel; RATHORE, Anirudh; TAN, Chenhao. Evaluating and characterizing human rationales. arXiv preprint arXiv:2010.04736, 2020.

[5] KOVÁCS, Ádám, et al. POTATO: exPlainable infOrmation exTrAcTion framewOrk. arXiv preprint arXiv:2201.13230, 2022.