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
This presentation contains profane words.
Outline

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

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

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

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

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

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“

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)

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

Influence of the rationales to the prediction

**Interpretation**: How accurately it reflects the true reasoning process of the model.

Two metrics:

- \( m(xi) \) is the probability that sentence \( xi \) is classified offensive.
- \( m(ri) \) is the probability that the predicted rationales \( ri \) alone are classified offensive.
- \( m(xi\setminus ri) \) is the sentence with removed predicted rationales.

**Comprehensiveness**:

(Were all features needed to make a prediction?)

- \( = m(xi) – m(xi\setminus ri) \)
- 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

How to remove continuous rationales?

→ Remove top k rationales (threshold)

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

$$
\frac{1}{|B| + 1} \left( \sum_{k=0}^{|B|} m(x_i)_{j} - m(x_i \setminus r_{ik})_{j} \right)
$$

ERASER Output

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

Faithfulness
Comprehensiveness : 0.6083561550950038
Sufficiency 0.15281228368862493

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

ERASER skips calculation

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 continuous
- However, a potato rule matches either fully or not
- Single sentence faithfulness metrics are either 0 or 1 (Smoothed out by aggregation)
Format

- jsonl
- Slightly different formats for ground truth and prediction
- Text is not in the jsonl but in the docs folder
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": 0.9711454510688782,
    "normal": 0.004742590710520744,
    "offensive": 0.024111928418278694
  },
  "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": 0.005441979970782995,
    "offensive": 0.9660893678665161,
    "normal": 0.028468627482652664
  },
}
```

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

```python
import rationale_benchmark.metrics as eraser
eraser.runEvaluation("None", # neutralclassname
data_dir=datadir, # data dir
split=testtrainorval, # split
results=pathtopredictions, # results
score_file=datadir+/erasure_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:**
```python
print_classification_report(df, stats)
print("-------------------")
matched_result = evaluator.match_features(df, features[target])
subgraphs = matched_result["Matched rule"]
labels = matched_result["Matched label"]
data_tsv_to_eraser(file)
prediction_to_eraser(file, subgraphs, labels, labels, labels, labels, target)
call_eraser("./hateplain", "val", ".//hateplain/val_prediction.json")
print("-------------------")
```
Applying the metrics to HateXplain

**Rationales**

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

**Dirty hack**

Add normal label in metrics.py

- 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

<table>
<thead>
<tr>
<th>Model</th>
<th>IOU F1</th>
<th>Token F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(BERT HXPlain)</td>
<td>(0.126)</td>
<td>(0.444)</td>
</tr>
<tr>
<td>(test.jsonl)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rules: sexism</td>
<td>0.279</td>
<td>0.165</td>
</tr>
<tr>
<td>val.tsv</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rules: homophobia</td>
<td>0.090</td>
<td>0.047</td>
</tr>
<tr>
<td>secondary_val.tsv</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

→ 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

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

\[ m(\xi/\rho) \]

\[ \xi/\rho = ??? \]

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)

5) Look at normalized ERASER metrics

Evaluate if needed, idea of a metric is to compare

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?
Measuring explainability in hate speech detection using the HateXplain dataset

Questions / Discussion / Thank you!

Sources:


