Intro & TLDR:

In order to improve our product we need to have:

- **Metrics** that reflect perception of our customer (i.e. human judgement)
- **Tools** for efficient data collection, data labelling, data curation and generation of synthetic data
- **Evaluation** procedures enabling us to compare various approaches

In this presentation we:

- Explain how be build our metrics, ending up in **two major metrics**:
  - **Match score** - indicating how good violations detected by our checkers (i.e. inference pipelines) are compared to ground-truth violations, considering highlighted text, violated rule, category, explanation and suggested correction
  - **F1 score** - indicating how many violations we detected were correct (and not hallucinated)
- Provide details of the evaluation split and elementary description of the evaluated checkers

How do we determine if there is a match 1

1. Calculate the ***text overlap***

- Overlap between the characters forming the ***ground-truth context*** and ***predicted context*** of violation
- ***text overlap*** = **IoU** (**Intersection over Union)** (vel. Jaccard similarity)

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Ground-truth: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Prediction: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Intersection: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Union: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

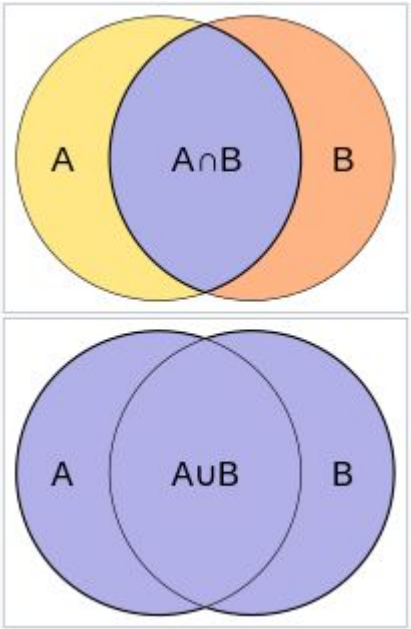


Image source: wikipedia

Fundamental metrics

- **True positive (**a.k.a. **match)**: we found a match between *ground-truth* and *predicted* violation
- **False negative:** violation that we missed
- **False positive:** sentence we flagged as a violation that is not present in ground-truth list of violations
- (**True negative**: not relevant to our case, we don't track this)

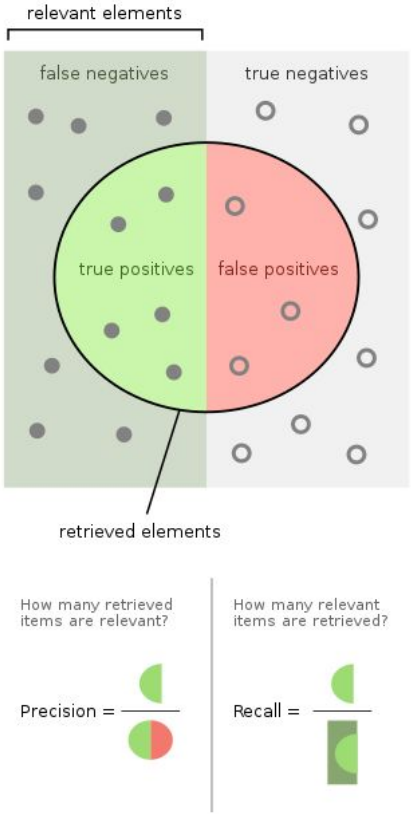
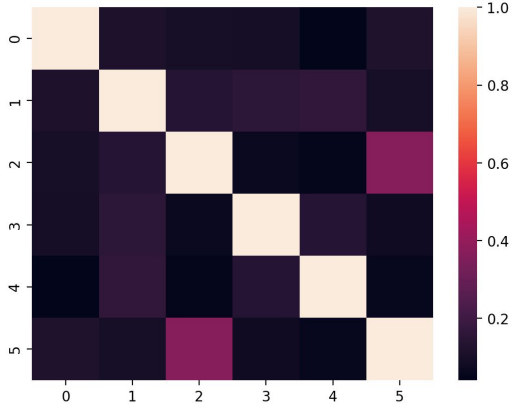


Image source: [wikipedia]

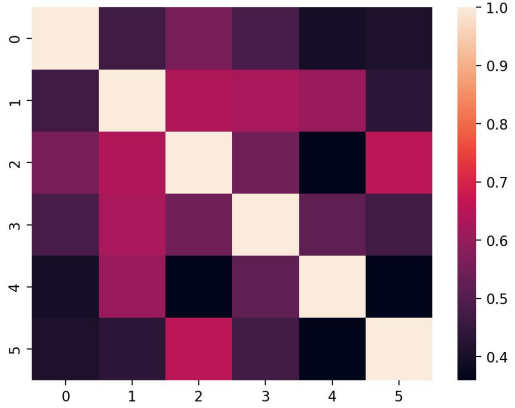
How do we determine if there is a match 2

2. Calculate the ***rule similarity***

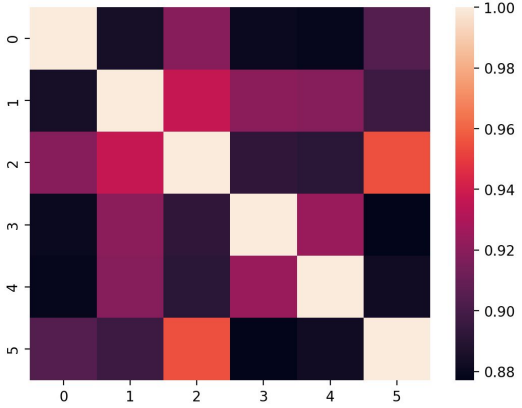
- Semantic similarity between the ***ground-truth rule*** and ***predicted rule***
- Using **sentence transformer**



string-based, Jaccard similarity (IoU)



sentence transformer, cosine similarity
(model: stsb-roberta-large)



OpenAI embeddings, cosine similarity
(model: text-embeddings-ada2)

How do we determine if there is a match 3

3. Building a matrix of **potential matches**

- Compare every *predicted* violation with every *ground-truth* violation
- Calculate the **weighted match score** defined as:
  - **weighted match score:** 0.5 * ***text overlap*** + 0.5 * ***rule similarity***

(Note: the weights are hyperparameters)

4. Finding a **match (**a.k.a. standard match, **std_match**)

- Core assumption: one *predicted* violation can match only one *ground-truth*, and vice versa
- Get the potential match with the **highest weighted match score** and check if:
  - ***text overlap*** must pass the threshold **> 0** (i.e. overlap of at least **one character**)
  - ***rule similarity*** must pass the threshold **> 0.01** (i.e. **1%** similarity in the embedding space)
  - **weighted match score** must pass the threshold **> 0.5 (new!)**
- If all check passed => it is a match

(Note: thresholds are hyperparameters too!)

Derived Metrics (yet still fundamental)

- **Precision**: how many violations in all **predicted violations** were **matches**
  (ideal: all predictions were matches, ratio = 1)

- **Recall**: how many **matches** from the **ground-truth violations** were found
  (ideal: every gt violation was matched, ratio = 1)

- **F1 score**: “the harmonic mean” of the precision and recall

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{2\mathrm{tp}}{2\mathrm{tp} + \mathrm{fp} + \mathrm{fn}}$$

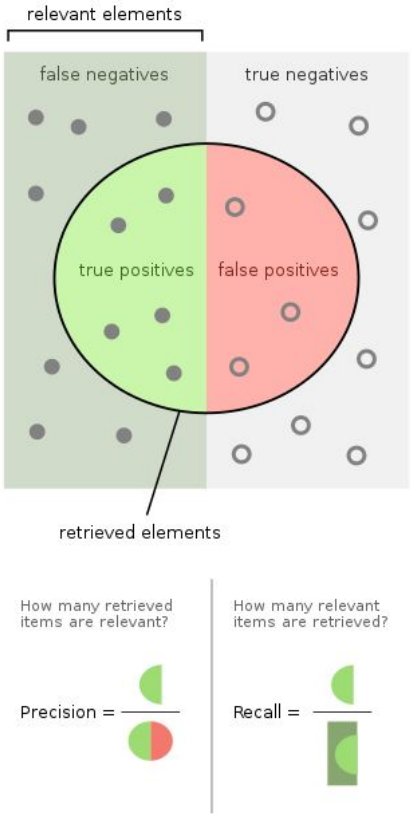


Image source: [wikipedia]

“Human-aligned” (ha_) metrics

- Take ***category match*** into consideration
  - Simple class matching (exact match)

- Take ***explanation match*** into consideration
  - Semantic similarity (Sentence Transformer)

- Take ***correction match*** into consideration
  - Semantic similarity (Sentence Transformer)

- Calculate ***ha_match*** based on **weighted match score** between all scores
  - 0.3 * ***text overlap*** +
    0.3 * ***rule similarity*** +
    0.1 * ***category match*** +
    0.2 * ***explanation match*** +
    0.1 * ***correction match***

- ***ha_false_positives***, ***ha_false_negatives*** -> ***ha_precision***/***ha_recall*** -> ***ha_f1***
  scores calculated in the same way as previously

Metrics: summary 1

Our evaluation enables to calculate **two sets of metrics**

- **Fundamental** and **derived metrics** are **the same in both cases**, but
- They **differ** on how we determine if/whether there is a **match**

For **std_match** we have the following hyperparameters:

- **Weights**:
  - text overlap = 0.5
  - rule similarity = 0.5
- **Thresholds**:
  - RULE_SIMILARITY_THRESHOLD = 1e-2 # 0.01, i.e. 1% similarity in the embedding space.
  - OVERLAP_THRESHOLD = 0 # IoU > 0, i.e. overlap of at least one character
  - MATCH_THRESHOLD = 0.5 # Threshold for considering a violation as a match.

Metrics: summary 2

For **ha_match** we have the following hyperparameters:

- **Weights**:
  - text overlap = 0.3
  - rule similarity = 0.3
  - category match = 0.1
  - explanation match = 0.2
  - correction match = 0.1

- **Thresholds**:
  - RULE_SIMILARITY_THRESHOLD = 1e-2 # 0.01, i.e. 1% similarity in the embedding space.
  - OVERLAP_THRESHOLD = 0 # IoU > 0, i.e. overlap of at least one character.
  - MATCH_THRESHOLD = 0.5 # Threshold for considering a violation as a match.

- => **Those parameters need to be *calibrated* to reflect our "human judgements" (future work)**

Data split and checkers

Data (used for evaluation) (split@version)

- **jaegermeister_curated_split@0.3.2**
  - Short texts (each text is a single paragraph with 1-2 sentences)
  - Collected from Jaegermeister publicly available materials, labelled & curated in-house
  - Number of samples: 7
  - Number of violations: 13

Checkers (pipelines for violation detection) (checker@version|model)

- **prompt-baseline@1.1.0|gpt-3.5-turbo-0613**
  - Naive checker implementation, with a simple prompt (“given rules find violations”)
- **zeroshot-index@2.2.5|gpt-3.5-turbo-0613**
  - Our first checker pipeline, implemented zero-shot prompt, detects violations and their categories
- **brute-context-classifier@1.7.5|gpt-3.5-turbo-0613**
  - One of the checkers we developed, now used mostly for data labelling due to high sensitivity
- **self-consistent-explainer@2.1.3|gpt-3.5-turbo-0613**
  - Our latest checker pipeline, implementing complex neuro-symbolic reasoning, with several reasoning steps, including voting and self-consistency checks