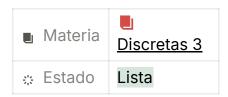
Module Design - Content Classification (DFAs)



File/Module

src/moderation/content_classification_dfa.py

Goal

Decide three booleans—hate, offensive, spam—using DFAs over a tiny alphabet.

Data model

@dataclass ClassificationReport

```
hate: bool offensive: bool spam: bool details: dict # {"tokens":[...], "symbols":[...], "counts":{"links":int,"hashtag s":int}}
```

Public API

classify(text: str) → ClassificationReport

Input: raw post string

Process:

- 1. Call preprocessing (extract_all) if available; else fallback tokenizer.
- 2. Map each token → symbol in {HATE, OFFENSIVE, LINK, HASHTAG, OTHER} via categorize.
- 3. Run three DFAs:
 - Hate DFA: accept if any HATE seen.
 - Offensive DFA: accept if any OFFENSIVE seen.

• Spam DFA: accept if ≥2 LINK or ≥3 HASHTAG.

Output: ClassificationReport

categorize(token: str) -> str

- Keep URLs/hashtags as is.
- Strip leading/trailing punctuation for keyword checks ("idiot!" → "idiot").
- Compare against sets.

Config knobs (top of module)

```
HATE_KEYWORDS = {"slur1","slur2"} # classroom placeholders 
OFFENSIVE_KEYWORDS = {"stupid","idiot"} 
MAX_LINKS_FOR_SAFE = 1 # 2+ \Rightarrow spam 
MAX_HASHTAGS_FOR_SAFE = 2 # 3+ \Rightarrow spam
```

Example

```
classify("go http://a.com http://b.com #wow")
# → hate=False, offensive=False, spam=True, details=...
```