






Module Desing - Content Transformation (FST)

 Materia	 <u>Discretas 3</u>
 Estado	Pendiente

File/Module:

`src/moderation/content_transformation_fst.py`

Goal:

Mask flagged words and emit suggestions.

Data model

@dataclass TransformResult

```
transformed_text: str      # final string with "****" replacements
masked_tokens: list[str]   # original tokens that got masked
suggestions: list[str]     # e.g., "Warning: offensive language detected."
categories: list[str]      # ["OTHER","OFFENSIVE","LINK",...]
original_tokens: list[str] # tokens before masking
```

Public API

transform(post: str) -> TransformResult

Input: raw post string

Process:

1. Prefer `classify(post)` to reuse tokens/symbols and spam/hate/offensive booleans.
2. For each token:
 - If symbol in `{HATE, OFFENSIVE}` → output `**` and record in `masked_tokens`.
 - Otherwise → keep token.
3. Build `suggestions` :
 - If hate → `"Warning: hate speech detected."`
 - If offensive → `"Warning: offensive language detected."`
 - If spam (2+ links or 3+ hashtags) → `"Notice: looks like spam (too many links/hashtags)."`

Output: `TransformResult`

Example

```
transform("You are IDIOT! visit http://a.com #wow")
# → transformed_text="you are *** visit http://a.com #wow"
#  masked_tokens=["idiot!"], suggestions=["Warning: offensive language detected."], ...
```