ChatGPT

Revisione automatica del glossario dei tratti

Questo documento sintetizza l'analisi automatica del file `data/core/traits/glossary.json` condotta dallo script `trait_review.py`. Lo script esegue le seguenti operazioni:

1. **Parsing del glossario**: carica tutti i tratti definiti nella sezione `traits` del JSON.
2. **Raggruppamento per prefisso**: per ogni `trait_id`, estrae il prefisso (ad esempio "antenne", "artigli", "branchie" ecc.) e conta quante varianti esistono per ciascun prefisso. Il risultato è salvato in `trait_categories.csv`.
3. **Rilevamento anomalie**: applica regole euristiche per individuare casi anomali:
4. label troppo descrittive (frasi lunghe, presenza di virgole o ellissi);
5. placeholder espliciti (per ora `random` e `pathfinder`);
6. traduzioni mancanti (label inglese identica a quella italiana o al `trait_id`);
7. mismatch di traduzione (grande differenza nella lunghezza delle due etichette);
8. refusi noti come "sghiaccio";
9. espressioni colloquiali nei nomi italiani. Le anomalie sono esportate in `trait_anomalies_auto.csv` con il tipo di problema e una nota esplicativa.
10. **Ricerca di duplicati**: segnala etichette italiane o inglesi uguali assegnate a più `trait_id`. Questi duplicati sono riportati in `trait_duplicates.csv`.

Come interpretare i risultati

- **trait_categories.csv**: mostra quante varianti esistono per ciascun prefisso. I prefissi con valori elevati indicano famiglie di tratti molto estese (es. antenne, branchie, artigli). È utile verificare che tutte le varianti abbiano descrizioni distinte e coerenti.
- **trait_anomalies_auto.csv**: elenca i tratti che infrangono le regole di consistenza. Questa tabella dovrebbe essere rivista manualmente per decidere se correggere i nomi, completare le traduzioni o rimuovere eventuali placeholder.
- **trait_duplicates.csv**: identifica etichette duplicate. Se la stessa etichetta è associata a più `trait_id`, occorre verificare se si tratta di un vero duplicato o se le voci vanno differenziate.

Prossimi passi

1. **Eseguire lo script**: assicurarsi che `trait_review.py` sia nella cartella `scripts/` del repository. Dal root del progetto, eseguire:

```
python scripts/trait_review.py --glossary data/core/traits/glossary.json --
outdir reports/trait_review
```

Questo creerà una cartella `reports/trait_review` con i file CSV.

1. **Analizzare i CSV**: importare `trait_categories.csv`, `trait_anomalies_auto.csv` e `trait_duplicates.csv` in un foglio di calcolo o in un software di analisi per esaminare i risultati.

2. **Applicare le correzioni**: sulla base dei report, aggiornare il `glossary.json` correggendo i nomi anomali, completando le traduzioni e armonizzando le serie di tratti dove necessario.

3. **Aggiornare la documentazione**: dopo la revisione del glossario, aggiornare i documenti di riferimento (es. `docs/catalog/trait_reference.md`) e rigenerare eventuali report di copertura.

Questo processo automatizzato facilita una revisione completa del glossario dei tratti, garantendo coerenza terminologica e qualità dei dati.
