PSEUDO-CODIFICA BUILD_ARFF

Nella pseudocodifica **vengono omessi**:

- creazione di un file contenente l'elenco delle features
- inizializzazione e copia delle features in un dizionario ordinato
- normalizzazione delle features
- scrittura del file arff ed interrogazioni spotlight / sparql
- altri dettagli implementativi

build_arff

init_arff // genera il file arff con l'elenco degli attributi (features + classe)

for each *t* in *topics*

write_arff_from_xml(t) // 1. legge il file xml contenente i post della sezione
// 2. inizializza il post
// 3. estrae le uri da spotlight
// 4. richiede le relazioni dbo da sparql
// 5. calcola e normalizza le features

write_arff_from_xml(topic)

for each *element* in *topic*

text ← *element.read_body*
post ← *new_post(text,topic)* // inizializza il post
post.init_features // genera l'elenco di features
write_to_arff(post) // scrive la riga corrispondente al post nel file arff

(post).init_features

uris ← *get_uris* // interroga spotlight per ottenere la lista di uri
rf ← *get_r_features(uris)* // dizionario delle feature ottenuto dalle relazioni
kf ← *get_k_features(uris)* // dizionario delle feature ottenuto dalle keywords
of ← *get_o_features(uris)* // altre features (numero di uri e lunghezza del testo)
normalize_features // normalizza i valori delle feature in [0,1]

get_r_features(uris)

relationships_features ← *empty_dictionary*

for each *u* in *uris*

    relationships_count ← *empty_dictionary*
    relationships_count.put(anime_relationships_count)
    relationships_count.put(beer_relationships_count)
    relationships_count.put(fitness_relationships_count)

    relationships_features.add(relarionships_count)

return relationships_features

get_k_features(uris, keywords)

keyword_features ← *empty_dictionary*
string_blob ← *empty_list*

for each *u* in *uris*

    string_blob.addAll(relevant_resources(u))

for each *k* in *keywords*

    count ← *keywords_count(string_blob, k)*
    key ← *keyword.superGroup(k)* // insieme a cui appartiene la parola chiave
    keyword_features.sum(key, count)

return keyword_features