package com.twitter.search.common.relevance.scorers;

import com.twitter.search.common.relevance.classifiers.TweetClassifier;

import com.twitter.search.common.relevance.entities.TwitterMessage;

/\*\*

\* Interface to compute feature scores for a single @TwitterMessage

\* object, or a group of them, after they have been processed by

\* feature classifiers.

\*

\* Intentionally kept Scorers separate from Classifiers, since they

\* may be run at different stages and in different batching manners.

\* Convenience methods are provided to run classification and scoring

\* in one call.

\*/

public abstract class TweetScorer {

/\*\*

\* Compute and store feature score in TwitterMessage based on its

\* TweetFeatures.

\*

\* @param tweet tweet message to compute and store score to.

\*/

public abstract void scoreTweet(final TwitterMessage tweet);

/\*\*

\* Score a group of TwitterMessages based on their corresponding TweetFeatures

\* and store feature scores in TwitterMessages.

\*

\* This default implementation just iterates through the map and scores each

\* individual tweet. Batching for better performance, if applicable, can be implemented by

\* concrete subclasses.

\*

\* @param tweets TwitterMessages to score.

\*/

public void scoreTweets(Iterable<TwitterMessage> tweets) {

for (TwitterMessage tweet: tweets) {

scoreTweet(tweet);

}

}

/\*\*

\* Convenience method.

\* Classify tweet using the specified list of classifiers, then compute score.

\*

\* @param classifier list of classifiers to use for classification.

\* @param tweet tweet to classify and score

\*/

public void classifyAndScoreTweet(TweetClassifier classifier, TwitterMessage tweet) {

classifier.classifyTweet(tweet);

scoreTweet(tweet);

}

/\*\*

\* Convenience method.

\* Classify tweets using the specified list of classifiers, then compute score.

\*

\* @param classifier classifier to use for classification.

\* @param tweets tweets to classify and score

\*/

public void classifyAndScoreTweets(TweetClassifier classifier, Iterable<TwitterMessage> tweets) {

for (TwitterMessage tweet: tweets) {

classifyAndScoreTweet(classifier, tweet);

}

}

}