package com.twitter.search.ingester.model;

import com.twitter.common.text.token.TokenizedCharSequenceStream;

import com.twitter.common.text.token.attribute.CharSequenceTermAttribute;

import com.twitter.search.common.relevance.text.VisibleTokenRatioNormalizer;

public class VisibleTokenRatioUtil {

private static final int TOKEN\_DEMARCATION = 140;

private static final VisibleTokenRatioNormalizer NORMALIZER =

VisibleTokenRatioNormalizer.createInstance();

/\*\*

\* Take the number of visible tokens and divide by number of total tokens to get the

\* visible token percentage (pretending 140 chars is visible as that is old typical tweet

\* size). Then normalize it down to 4 bits(round it basically)

\*/

public int extractAndNormalizeTokenPercentage(TokenizedCharSequenceStream tokenSeqStream) {

CharSequenceTermAttribute attr = tokenSeqStream.addAttribute(CharSequenceTermAttribute.class);

int totalTokens = 0;

int numTokensBelowThreshold = 0;

while (tokenSeqStream.incrementToken()) {

totalTokens++;

int offset = attr.getOffset();

if (offset <= TOKEN\_DEMARCATION) {

numTokensBelowThreshold++;

}

}

double percent;

if (totalTokens > 0) {

percent = numTokensBelowThreshold / (double) totalTokens;

} else {

percent = 1;

}

return NORMALIZER.normalize(percent);

}

}