package com.twitter.tweetypie.matching

import com.twitter.common.text.pipeline.TwitterLanguageIdentifier

import com.twitter.common\_internal.text.version.PenguinVersion

import java.util.Locale

object TweetTokenizer extends Tokenizer {

type LocalePicking = Option[Locale] => Tokenizer

/\*\*

\* Get a Tokenizer-producing function that uses the supplied locale

\* to select an appropriate Tokenizer.

\*/

def localePicking: LocalePicking = {

case None => TweetTokenizer

case Some(locale) => Tokenizer.forLocale(locale)

}

private[this] val tweetLangIdentifier =

(new TwitterLanguageIdentifier.Builder).buildForTweet()

/\*\*

\* Get a Tokenizer that performs Tweet language detection, and uses

\* that result to tokenize the text. If you already know the locale of

\* the tweet text, use `Tokenizer.get`, because it's much

\* cheaper.

\*/

def get(version: PenguinVersion): Tokenizer =

new Tokenizer {

override def tokenize(text: String): TokenSequence = {

val locale = tweetLangIdentifier.identify(text).getLocale

Tokenizer.get(locale, version).tokenize(text)

}

}

private[this] val Default = get(Tokenizer.DefaultPenguinVersion)

/\*\*

\* Tokenize the given text using Tweet language detection and

\* `Tokenizer.DefaultPenguinVersion`. Prefer `Tokenizer.forLocale` if

\* you already know the language of the text.

\*/

override def tokenize(tweetText: String): TokenSequence =

Default.tokenize(tweetText)

}