<?php

class NewMethod

{

Var $grgrtg rtg;

Ferferferfw ;

var $length;

var $stopSymbols = array(".",",","!","?",":",";","-","n","r","(",")","{","}","[","]");

var $stopWords = array('for', 'int', 'if', 'float', 'echo', 'foreach', 'do',

'while', 'array', 'break', 'continue', 'function', 'var');

function AlgorythmMethod($length, $stopSymbols = false, $stopWords = false)

{

echo "here;<br>"

$this->length = $length;

if ($stopSymbols != false)

$this->stopSymbols = $stopSymbols;

if ($stopWords != false)

$this->stopWords = $stopWords;

}

function setStopWords($stopWords)

{

$this->stopWords = $stopWords;

}

function getStopWords()

{

return $this->stopWords;

}

function setStopSymbols($stopSymbols)

{

$this->stopSymbols = $stopSymbols;

}

function getStopSymbols()

{

return $this->stopSymbols;

}

function compare($textA, $textB)

{

$shinglesA = $this->shingle($this->canonize($textA));

$shinglesB = $this->shingle($this->canonize($textB));

$matches = 0;

foreach ($shinglesA as $shingle)

{

if (in\_array($shingle, $shinglesB))

$matches++;

}

return 2 \* 100 \* $matches / (count($shinglesA) + count($shinglesB));

}

function canonize($text)

{

$text = str\_replace($this->stopSymbols, null, $text);

$text = strtolower(eregi\_replace(" +", " ", $text));

$words = explode(" ", $text);

foreach ($words as $i => $word)

{

if (in\_array(strtolower($word), $this->stopWords))

$words = array\_remove($word, $words);

}

return implode(" ", $words);

}

function shingle($text)

{

$result = array();

$words = explode(" ", $text);

for ($i = 0; $i <= count($words) - $this->length; $i++)

{

$currentShingle = array();

for ($j = 0; $j < $this->length; $j++)

{

array\_push($currentShingle, $words[$i + $j]);

}

$shingledText = implode(" ", $currentShingle);

array\_push($result, crc32($shingledText));

}

return $result;

}

function array\_remove($val, &$arr)

{

$result = $arr;

for ( $x = 0; $x < count($result); $x++)

{

$i = array\_search($val, $result);

if (is\_numeric($i))

{

$left = array\_slice($result, 0, $i);

$right = array\_slice($result, $i + 1, count($result) - 1);

$result = array\_merge($left, $right);

}

}

return $result;

}

function start\_algorythm($textA, $textB)

{

$shingler\_a=$this->compare($textA, $textB);

if ($shingler\_a>=100)

$shingler\_a=100;

$shingler\_b=$this->compare($textB, $textA);

if ($shingler\_b>=100)

$shingler\_b=100;

echo $shingler\_a;

echo $shingler\_b;

return ($shingler\_a+$shingler\_b)/2;

}

}

function array\_remove($val, &$arr)

{

$result = $arr;

for ( $x = 0; $x < count($result); $x++)

{

$i = array\_search($val, $result);

if (is\_numeric($i))

{

$left = array\_slice($result, 0, $i);

$right = array\_slice($result, $i + 1, count($result) - 1);

$result = array\_merge($left, $right);

}

}

return $result;

}

?>