A Method for Measuring Keywords Similarity by Applying Jaccard's, N-Gram and Vector Space

Jatsada Singthongchai and Suphakit Niwattanakul
School of Information Technology,Suranaree University of Technology,Nakhon Ratchasima, Thailand.
E-mail: jatsada_007@hotmail.com, suphakit@sut.ac.th

***Abstract*—Obviously, searching engines today have not sufficiently fulfilled the needs of users. Most of these searching engines are functioned using keywords query which are identical and relevant to conceptual search. This means that matched or successful searching results depend on how a user spells the keywords. Thus, the method for measuring keywords similarity between keywords query and index words is very crucial. This research focuses on keywords search. We have designed method for measuring keywords similarity with Jaccard's, N-Gram, Vector space, Average (JNVA) and Jaccard's, N-Gram, Length, Average (JNLA) by using hybrid method; a combination of Jaccard's , N-Gram and Vector Space to make Keywords search practical. These methods are evaluated by three criterions (precision, recall, and F-measure). The result reveals that the method for measuring keywords similarity with the application of JNVA and JNLA can successfully predict the similarity between keywords query with index words. These methods can be applied in order to develop searching engines performance especially semantic search.**

***Index Terms*—keywords similarity, Jaccard's, N-Gram, vector space**

I. Introduction

Recently, internet access has played important role to our daily life in communicating and spreading information. Thus, internet is the mainstream source of information. When popularity of internet usage increases, data also increases correspondingly. Sometimes, searching becomes difficult so there are many searching engines available in the cyber-world today to facilitate internet users. However, the most important factor for an effective and successful searching is keyword. For example, the most efficient keyword that user has input will link to the relate document he requires. This has proved that keyword is very important in searching activity [1]-[4] explain that the ultimate goal of information searching activity is to find any document that users require which relate to keyword input. [5] also explains that searching procedure stats with matching input keyword to word index.

Three obvious common problems occurred when internet searching activity performed are 1) Inputting keywords that doesn't correspond to conditions of searching engine index. 2) Inputting too short keywords that confuse the engine. 3) Inputting keywords with wrong spelling. Solving these problems needs effective keyword similarity coefficient. The most approved methods in the business today are Jaccard's N-Gram and Vector Space. Each of these methods has weakness and strength in measuring similarity. [6] proposes keyword similarity measurement with Jaccard's Similarity Coefficient to compare likeliness between sets of data. They developed measurement engine from Jaccard's Similarity Coefficient by applying Prolog. The result of their study reveals that Jaccard's Similarity Coefficient method can better identify similarity between keywords and words contained in the index when characters of word are compared. The engine will compute and recognize the original word then displays the result precisely no matter that characters of the keyword are typed in the wrong order. However, any misspelling keyword cannot be reckoned as a correct word if there are more than one characters typed incorrectly.

So, we have designed method for measuring keywords similarity with Jaccard's, N-Gram, Vector space, Average (JNVA) and Jaccard's, N-Gram, Length, Average (JNLA) by using hybrid method; a combination of Jaccard's , N-Gram and Vector Space to make Keywords search practical.

II. Related Theories and Studies

A. Jaccard's

Jaccard Similarity Coefficient is a parameter used to compare characteristic similarity between sets of information. Similarity measurement of Jaccard's between two example sets is a quotient of sharing characteristic number divided by all characteristic number displayed in the first equation.

$$Jaccard = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

A = set of characters that doesn't occur in word 1
B = set of characters that doesn't occur in word 2

Jaccard's can be applied in analyzing meaning of a word by deducing the word as a set of characters. This relates to the study of [5] in which he creates word searching procedure by matching a word that a user requires to words contained in the index. If the word input matches to the ones in the index, the input word will be the main in the search. However, if the word input

Manuscript received June 20, 2013; revised August 29, 2013.

doi: 10.12720/lnit.1.4.159-164

doesn't match to the one in the index, the input word will be computed through keyword similarity measurement by Jaccard Similarity Function. The result is quite acceptable.

B. N-Gram

N-Gram is a model which is used to calculate probability of character sequence that occurs as a word or probability of word sequence that occurs as a sentence. Probability of character can be estimated from source of data. N-Gram size are varied depending on how large programmer would set. It can be from 1 to (n). In this N-Gram model, the length of character and word sequence are different (2-3 Gram and 4 Gram) as displayed in the second equation.

$$Ngram(2, X) = \{x_0x_1, x_1x_2, x_2x_3, ...x_{n-1}x_n\} \quad (2)$$

$$Ngram(3, X) = \{x_0x_1x_2, x_1x_2x_3, x_2x_3x_4, ..., x_{n-2}x_{n-1}x_n\}$$

N-Gram model is a popular statistical method that is used today and suitable for analyzing language. It can recognize language. [7] propose a study about an information searching system developed by N-Gram that provides accurate information. The system is evaluated 78% accuracy. [8] proposes a study about recognizing Thai and English words by N-Gram. This study shows how to recognize single Thai words that are continually written next to others without spacing like words written in English. N-Gram is applied in order to split single Thai words from others by calculating number of characters. However, only N-Gram cannot fully recognize all of single Thai words from others in phrases, sentences or passages without Grammar rules application.

C. Vector Space

Vector Space model is developed by [4]. Vector Space is an algebra that presents document identified by [4] and [9] explain that documents and keywords are represented in forms of t-dimensional Space Vector of word weighting. For example, word index is normally created by Vector Space model in verifying and ordering data in information searching system that displayed in the third equation.

$$Vectorspace = \left(\frac{V_A}{len(A)}\right).\left(\frac{V_B}{len(B)}\right) \quad (3)$$

V_A = Vector of number of characters in word 1
V_B = Vector of number of characters in word 2
A = set of all characters in word 1
B = set of all characters in word 2

[10] explains that principle of Vector Space model is a representation of document and inquiry. Displayed and input data are imported by Vector of Terms. Each core of Vector is "terms" that occurred in the document. [11] has applied Vector Space model and principle of word weighting in developing automatic answering system. Then, she measures similarity by using Euclidian Distance. Euclidian Distance is a measurement between distances of data needed to compare similarity. The limitation of this measuring method is sensitiveness between distance, size of objects that are different, and correlation.

III. Methodology

A. Word Interrelation Data Preparatory

There are two parts of data in this research which are 1) set of words in the database ("words" here also include phrases), that are grammatically correct, taken from 300 AGROVOC Thesaurus of Food and Agriculture Organization of the United Nations (FAO) and Thai AGROVOC Thesaurus words list, and 2) set of words that is grammatically wrong taken from 3 misspelling cases done by users that are 100 misspelling words, 100 incomplete spelling words, and 100 over spelling words specified by researchers. The keywords examples are displayed in Table I.

TABLE I. Example of Words that Appear in the Index and Keywords Query.

Keywords Query	Index words
correctly spelled words	Cane
Rice, Crops, Cane, Sugarcane,Manioc,Cassava,	Sugarcane
	Manioc
	Cassava
misspelling	Mangoes
Rioc, Crips , Cade, sudarcane,madioc, cadsava	Tangerines
	Pineapples
	Soybeans
Incomplete spelling	Maize
Ric, Crps, Cae, sugarcane, Maioc, cassava	Rice
	Barley
	Crops
over spelling	Cereals
Ricee, Cropss, Canne, Suggarcane, Mannioc, Casssava	Plants
	Soybeans
	Limes

B. Keyword Similarity Measurement Design

We have studied the three popular methods of keyword similarity measurement (Jaccard's, N-Gram, and Vector Space) to find out advantage and disadvantage of each method. So, we have designed method for measuring keywords similarity with JNVA and JNLA by using hybrid method; a combination of Jaccard's, N-Gram and Vector Space to make Keywords search practical as displayed in the fourth and the fifth equation.

- Jaccard's N-Gram Vector space Average (JNVA) is created by combining 3 strong points of the three methods which expand limitation of similarity measurement.

$$JNVA = \frac{Jaccard + Ngram2 + Vectorspace}{3} \quad (4)$$

- Jaccard's N-Gram Length Average (JNLA) presents coefficient of character length supports when typing mistakes are compared.

$$JNLA = \frac{Jaccard + Ngram2 + hlengt}{3}$$

$$hlengt = \exp\left[\frac{-abs(len(A) - len(B))}{len(A \cup B)}\right] \quad (5)$$

A = set of characters that doesn't occur in word 1
B = set of characters that doesn't occur in word 2

C. Algorithms and Coding Design

We have design algorithms to illustrate keywords similarity measurement methods that are already designed. Algorithms are set to Pseudo code with PHP language. The result can be displayed via web browser. The methods are evaluated by three criterions (precision, recall, and F-measure). [12]

- *Precision* can be evaluates by the following formula.

$$precision = \frac{TP}{TP + PP} \times 100\% \quad (6)$$

TP (True Positive) = keywords presented and accepted by experts
FP (False Positive) = keywords presented but not accepted by experts

- *Recall* can be evaluates by the following formula.

$$recall = \frac{TP}{TP + FN} \times 100\% \quad (7)$$

TP (True Positive) = keywords presented and accepted by experts
FN (False Negative) = keywords not presented and not accepted by experts

- *F-measure* can be evaluates by the following formula.

$$F = \left(2 \times precision \times recall\right) / \left(precision + recall\right) \quad (8)$$

IV. Results and Discussion

From the keywords similarity measurement with JNVA and JNLA, we would like to show the two designs and experiment procedure as follow;

A. Algorithms of Method for Measuring Keywords Similarity with JNVA.

Inputting string1 : str1
Inputting string 2 : str2
Calculating unique string1 : ustr1
Calculating unique string2 : ustr2
Jaccard(ustr1,ustr2) :
  Calculating union of ustr1 and ustr2 : unstr
  Calculating intersection of ustr1 and ustr2 : instr
  Calculating result of Jaccard : Jacstr = |unstr|/|instr|
Ngram2(str1,str2) :
  Creating data set 1 from str1 with Ngram : ngm1
  Creating data set 2 from str2 with Ngram : ngm2
  Calculating unique ngm1: ungm 1
  Calculating unique ngm2: ungm 2
  Calculating result of Ngram2 : Ngmstr = Jaccard(ungm1,ungm2)
VectorSpace(ustr1,ustr2,str1,str2):
  Creating vector space data from ustr1 and ustr2 : vt
  Creating vector space data 1 from vt and str1 : vtstr1
  Creating vector space data 2 from vt and str2 : vtstr2
  Calculating unit vector 1 : uvtstr1 = vtstr1/Len(str1)
Calculating unit vector 2 : uvtstr2 = vtstr2/Len(str2)
  Calculating result of Vector Space : Vspstr = uvtstr1 uvtstr2
Calculating result of JNVA : JNVA = (Jacstr + Ngmstr + Vspstr)/3

From the displayed algorithms can be easily illustrated by a diagram as Fig. 1.

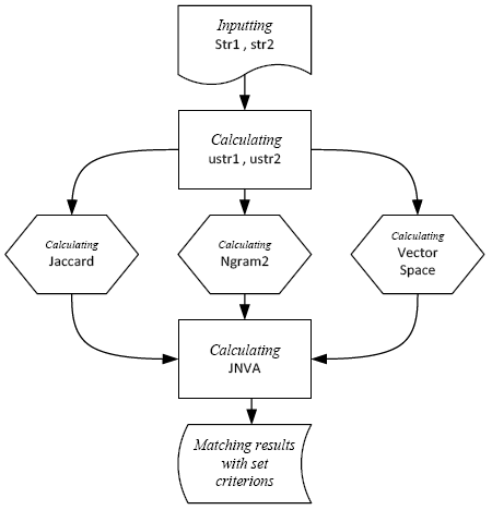


Figure 1. Display a diagram of method for measuring keywords similarity with JNVA.

B. Algorithms of Method for Measuring Keywords Similarity with JNLA.

Inputting string1 : str1
Inputting string 2 : str2
Calculating unique string1 : ustr1
Calculating unique string2 : ustr2
Jaccard(ustr1,ustr2) :
Calculating union of ustr1 and ustr2 : unstr
  Calculating intersection of ustr1 and ustr2 : instr
  Calculating Jaccard : Jacstr = |unstr|/|instr|
Ngram2(str1,str2) :
  Creating dataset 1 from str1 with Ngram : ngm1
  Creating dataset 2 from str2 with Ngram : ngm2
  Calculating unique ngm1: ungm 1
  Calculating unique ngm2: ungm 2
  Calculating result of Ngram2 : Ngmstr = Jaccard(ungm1,ungm2)
Length(ustr1,ustr2,str1,str2):
  Calculating differences of str1 and str2 : dfstr = abs(Len(str1)-Len(str2))
  Calculating intersection of ustr1 and ustr2 : instr
  Calculating result of Length Lenstr = EXP(-dfstr/Len(instr))
Calculating result of JNLA : JNLA = (Jacstr + Ngmstr + Lenstr)/3

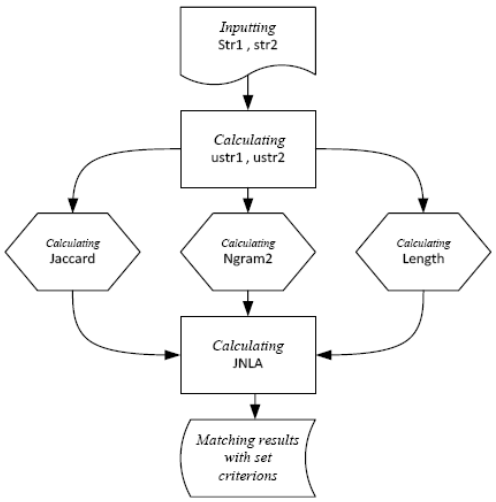


Figure 2. Display a diagram of method for measuring keywords similarity with JNLA.

From the displayed algorithms can be easily illustrated by a diagram as Fig. 2.

C. Code Example with PHP language

```php
function jaccard($str1,$str2)
{
    $astr1 = array_unique(mbStringToArray($str1));
    $astr2 = array_unique(mbStringToArray($str2));
    $union = array_unique(array_merge((array)$astr1, (array)$astr2));
    $intersect = array_intersect((array)$astr1, (array)$astr2);
    $ucount = count($union);
    $icount = count($intersect);
    return ($icount/$ucount);
}

function n2gram($str1,$str2)
{
    $astr1 = (mbStringToArray($str1));
    $astr2 = (mbStringToArray($str2));
    $stack = array();
    array_push($stack, " ".$astr1[0]);
    for($i=0;$i<count($astr1)-1;$i++)
    {
    array_push($stack, $astr1[$i].$astr1[$i+1]);
    }
    array_push($stack, $astr1[count($astr1)-1]." ");
    $astr1=array_unique($stack);
    $stack = array();
    array_push($stack, " ".$astr2[0]);
    for($i=0;$i<count($astr2)-1;$i++)
    {
    array_push($stack, $astr2[$i].$astr2[$i+1]);
    }
    array_push($stack, $astr2[count($astr2)-1]." ");
    $astr2=array_unique($stack);
    $union = array_unique(array_merge((array)$astr1, (array)$astr2));
    $intersect = array_intersect((array)$astr1, (array)$astr2);
    $ucount = count($union);
    $icount = count($intersect);
    return ($icount/$ucount);
}

function vectorspace($str1,$str2)
{
   $astr1 = (mbStringToArray($str1));
   $astr2 = (mbStringToArray($str2));
   $astr = array_unique(array_merge((array)$astr1, (array)$astr2));
   $sum = 0;
   for ($i = 0; $i < count($astr); $i++)
   {
       $count1 = 0;
       $count2 = 0;
       for ($j = 0; $j < count($astr1); $j++)
       {
           if($astr[$i]==$astr1[$j])
           {
               $count1++;
           }
       }
       for ($j = 0; $j < count($astr2); $j++)
       {
           if($astr[$i]==$astr2[$j])
           {
               $count2++;
           }
       }
       $sum = $sum + sqrt(($count1 / count($astr1)) * ($count2 / count($astr2)));
   }
   return $sum; }

function length($str1,$str2)
{
    $astr1 = (mbStringToArray($str1));
    $astr2 = (mbStringToArray($str2));
    $union = array_unique(array_merge((array)$astr1, (array)$astr2));
    $ucount = count($union);
    $icount = count($astr1)-count($astr2);
    return exp(-abs($icount)/$ucount);
}
```

```php
function JNVA($str1,$str2)
{
    $j = jaccard($str1,$str2);
    $n = n2gram($str1,$str2);
    $v = vectorspace($str1,$str2);
    $s = ($j+$n+$v)/3;

    return round($s,3);
}

    function JNLA($str1,$str2)
{
    $j = jaccard($str1,$str2);
    $n = n2gram($str1,$str2);
    $l = length($str1,$str2);
    $s = ($j+$n+$l)/3;

    return round($s,3);
}
```

Figure 3. Example of Function creation code in PHP Language

When each method is analyzed, the JNVA and JNLA designs are applied. After the application is finalized, functions of each method are created as displayed in Fig. 3. These functions are in form .inc files. These files can be accessed by using "*require_once("function.inc");" code* with .php files.

D. Testing Result of Method for Measuring Keywords Similarity with JNVA and JNLA.

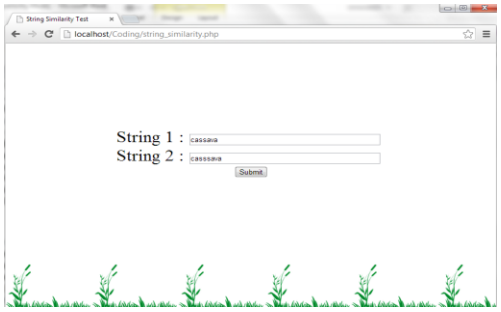
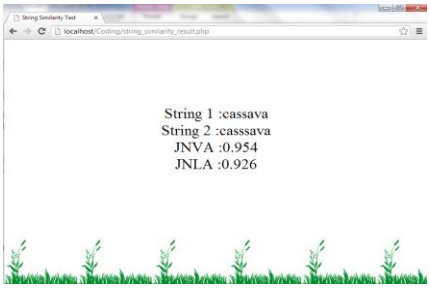
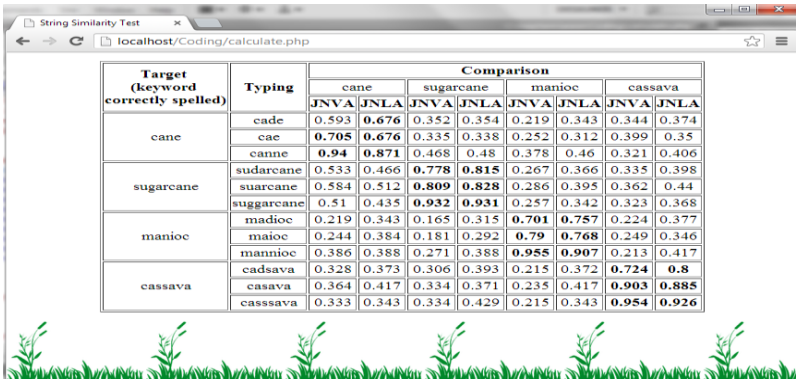


Figure 4. Displays Method for Measuring Keywords Similarity with JNVA and JNLA from keywords query.

From Fig. 4 we have input keywords by typing keywords query. It is displayed as the Fig. 5 and Table II below.





Target (keyword correctly spelled)	Typing	Comparison							
		cane		sugarcane		manioc		cassava	
		JNVA	JNLA	JNVA	JNLA	JNVA	JNLA	JNVA	JNLA
cane	cade	0.593	**0.676**	0.352	0.354	0.219	0.343	0.344	0.374
	cae	**0.705**	**0.676**	0.335	0.338	0.252	0.312	0.399	0.35
	canne	**0.94**	**0.871**	0.468	0.48	0.378	0.46	0.321	0.406
sugarcane	sudarcane	0.533	0.466	**0.778**	**0.815**	0.267	0.366	0.335	0.398
	suarcane	0.584	0.512	**0.809**	**0.828**	0.286	0.395	0.362	0.44
	suggarcane	0.51	0.435	**0.932**	**0.931**	0.257	0.342	0.323	0.368
manioc	madioc	0.219	0.343	0.165	0.315	**0.701**	**0.757**	0.224	0.377
	maioc	0.244	0.384	0.181	0.292	**0.79**	**0.768**	0.249	0.346
	mannioc	0.386	0.388	0.271	0.388	**0.955**	**0.907**	0.213	0.417
cassava	cadsava	0.328	0.373	0.306	0.393	0.215	0.372	**0.724**	**0.8**
	casava	0.364	0.417	0.334	0.371	0.235	0.417	**0.903**	**0.885**
	casssava	0.333	0.343	0.334	0.429	0.215	0.343	**0.954**	**0.926**

Figure 5. displays example result of each method

TABLE II. KEYWORDS SIMILARITY MEASURING EXAMPLE OF JNVA AND JNLA

Target (keyword correctly spelled)	Typing	JNVA		JNLA	
		Sugarcane*	Cassava*	Sugarcane*	Cassava*
Sugarcane*	sudarcane **	**0.778**	0.335	**0.815**	0.398
	suarcane ***	**0.809**	0.362	**0.828**	0.44
	Suggarcan****	**0.932**	0.323	**0.931**	0.368
Cassava*	cadsava **	0.306	**0.724**	0.393	**0.8**
	casava ***	0.334	**0.903**	0.371	**0.885**
	casssava ****	0.334	**0.954**	0.429	**0.926**

(*) means correctly spelled words (**) means misspelling
(***) means incomplete spelling (****) means over spelling

From Table II, it indicates that numbers of JNVA and JNLA similarity are zero to one. Zero signifies none of similarity. The results shows that JNVA and JNLA keywords similarity measurement value is high such as "Suggarcan" (****) with "Sugarcane" (*) indicates 0.932 and 0.931, and "casava"(***) with "Cassava"(*) indicates 0.903 and 0.885 respectively.

Then researchers count keywords similarity coefficient values that indicate more than 0.67 as selected words. On the other hand, the ones that are lower than 0.67 are counted as not selected words. Then, all values are calculated to identify keywords similarity measuring efficiency with criteria of precision, recall and F-measure explained above in the sixth, seventh and eighth equations and as displayed in Table III and Fig. 6.

TABLE III. KEYWORDS SIMILARITY MEASURING EXAMPLE OF JNVA AND JNLA

Method	Precision	Recall	measure-F
Jaccard's	100.00	95.55	97.41
N-Gram	100.00	57.77	69.27
Vector Space	89.92	100.00	93.89
JNVA	100.00	97.77	98.82
JNLA	100.00	97.77	98.75

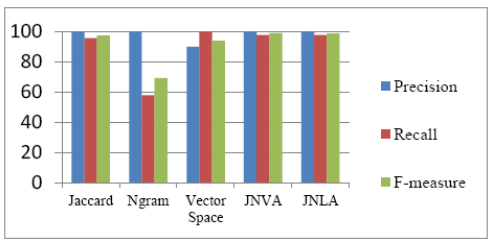


Figure 6. Result of keywords similarity measuring evaluation

From the result with the keywords similarity acceptance at the rate over 0.67, it indicates that JNVA keywords similarity measurement has 100.00 precision, 97.77 recall, and 98.82 F-measure, and JNLA keywords similarity measurement has 100.00 precision, 97.77 recall, and 98.75 F-measure. This can conclude that the designed methods (JNVA and JNLA) can better deliver keywords similarity measurement value prediction and are more stable than Jaccard's, N-Gram, and Vector space

V. CONCLUSION AND FURTHER STUDIES

This study has studied and developed ways to measure keywords similarity. In the progress of researching, it reveals that Jaccard's can freely analyze characters in each word especially when incomplete spelling and misspelling words encountered, Jaccard's can effectively measure and predict with high stability. However, when over spelling occurred, predictable accuracy decreases with low stability. N-Gram can effectively compare position of characters but there is an obvious high restriction of position because N-Gram just only expand limitation of data. Vector Space can effectively analyze number of characters but if wrong characters occurs in the word, there will be a mistake in calculation. Obviously, from the studies the three methods, it reveals that each of them has strength and weakness in its own way. Thus, this research has designed way to combine these three methods to achieve the most effective ways of identifying keywords similarity coefficient which are Jaccard's N-Gram Vector space Average (JNVA) and Jaccard's N-Gram Length Average (JNLA) to correct weakness of the three mentioned methods.

From the result of Method for Measuring Keywords Similarity with JNVA and JNLA, it reveals that the proposed methods are efficient in order to improve keywords query with correctness and precision. JNVA focuses on measuring number of alphabet in a keyword. However, JNLA focuses measuring number of all alphabet in each word contained in keyword. The application of these methods depends on purpose or case study. However, these methods can be applied in order to develop searching engines performance especially semantic search.

REFERENCES

[1] D. M. Christopher, R. Prabhakar, and S. Hinrich, *Introduction to Information Retrieval*, 1st ed. Cambridge, U.K.: Hardback, 2008, pp. 151-175.

[2] A. Lourdes and R. A. José, "Improving query expansion with stemming terms: A new genetic algorithm approach," in *Proc. 8th European Conf. Evolutionary computation in Combinatorial optimization*, 2008, pp. 182-193.

[3] K. Drabenstott and D. Vizine-Goetz, *Using Subject Headings for Online Retrieval: Theory, Practice and Potential*, 1st ed. U.K.: Emerald, 1994, pp. 132-218.

[4] G. Salton, *Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by Computer*, USA: Addison-Wesley Longman, 1989, pp. 189-295.

[5] S. Nivattanakul, "Access to knowledge based-on an ontology model," Ph.D. thesis, University of La Rochelle, 2008.

[6] S. Nivattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," in *Proc. International Muti Conf. Engineers and Computer Scientists*, Hong Kong, 2013, pp. 380-384.

[7] P. Sittichoke and N. Siranee, "Information retrieval system using N-Gram technique," in *Proc. 5th National Conf. Computing and Information Technology*, Bangkok, Thailand, 2009, pp. 307-312.

[8] A. Ekwonganan, "Identification of Thai and transliterated words by N-Gram models," thesis, Chulalongkorn University. Bangkok, 2005.

[9] J. H. Lee, "Combining multiple evidence from different properties of weighting schemes," in *Proc. 18th Annual. International Conf. Research and Development in Information Retrieval*, New York, 1995, pp.180-188.

[10] P. Phumphuong and C. Jaruskulchai, "Thai sentence boundary detection by using support vector machine," presented at 3rd International Symposium Conf. Communications and Information Technologies, Songkhla, Thailand, September 3-5, 2003.

[11] N. Hommuang, "The development questions answering system using a hybrid vector space mode," thesis, King Mongkut's University of Technology North Bangkok, 2007.

[12] D. Miao, Q. Duan, H. Zhang, and N. Jiao, "Rough set based hybrid algorithm for text classification," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9168-9174, 2009.

Jatsada Singthongchai is at the School of Information Technology, Suranaree University of Technology, Thailand. Currently he is lecturer at the school of Information Technology, Rajamangala University of Technology Isan Kalasin Campus, Thailand. He completed his MSc in Internet and Information Technology, from the Naresuan University, Thailand in 2004. His research areas include Semantic search, Internet Applications and Web Accessibility.

Suphakit Niwattanakul is at the school of Information Technology, Suranaree University of Technology, Thailand. He received his PhD in computer science from the University of La Rochelle, France in 2008. His current research is about Semantic Web technologies applied to information extraction and retrieval system.