An Automatic Approach to Generating Abstractive Summary for Thai Opinions

[1]Orawan Chaowalit, [2]Ohm Sornil

[1,2] *Department of Computer Science, National Institute of Development Administration, Bangkok, Thailand*
[*1]*orawan@su.ac.th*, [2]*osornil@as.nida.ac.th*

Abstract

With the advancement in the Internet technology, customers can easily share opinions on services and products in forms of reviews. There can be large amounts of reviews for popular products. Manually summarizing those reviews for important issues is a daunting task. Automatic opinion summarization is a solution to the problem. The task is more complicated for reviews written in Thai language. Thai words are written continuously without space, there is no symbol to identify the end of a sentence, and many reviews are written informally, thus accurate word identification and linguistic annotation cannot be relied upon. Text summarization can be classified into two categories which are extractive and abstractive summarization. In an extractive method, a summary is a set of actual sentences or phrases extracted from the reviews. Abstractive summarization does not output original sentences from the reviews but generates new sentences or phrases as a summary. This approach is more difficult and thus less popular than is the extractive approach. This research proposes a novel technique to generate abstractive summaries of customer reviews written in Thai language. The proposed technique, which consists of the local and the global models, is evaluated using actual reviews of fifty randomly selected products from a popular cosmetic website. The results show that the local model outperforms the other model and the two baseline methods both quantitatively and qualitatively.

Keywords: *Abstractive Opinion Summarization, Thai Language, Hopfield Network Algorithm*

1. Introduction

Nowadays consumers can buy products and services from vendors around the world. Companies need to secure and expand their market shares. Customer satisfaction is critical to maintaining customer base and developing as well as improving their products. Companies hire market researchers to survey customer satisfactions. Customers also want information from others to decide whether products are good or suitable for them. With the advent of the Internet, product reviews can be done through web boards, web blogs, companies' or third-party websites. This makes it easier and more convenient for companies to have necessary information to improve their products and for customers to have aids for their buying decisions.

The number of reviews has increased to over a hundred reviews per product or more if the product is popular, and the websites are well known and trusted by customers. Customer reviews are mostly unstructured, natural language texts. Some are long, some are short, some are grammatically incorrect, and some are short phrases. Reviews are usually about details and properties of products and are redundant in contents which indicate that those properties are crucial to customers. Due to the amount of reviews, it is difficult for readers to summarize main ideas of those opinions. An automatic opinion summarization is a solution to this problem by automatically summarizing core ideas of the entire set of reviews.

Text summarization, in generally, can be classified into two categories: extractive and abstractive summarization. In an extractive method [1,2], a summary is a set of actual sentences or phrases extracted from the reviews. Most research in this category focuses on texts relevant to the title or a topic of interest [1], and the output is a set of representative sentences from the original review texts, according to some criteria. Abstractive summarization [2] does not output original sentences from the reviews but generates new sentences or phrases as a summary. This approach is more difficult and thus less popular than is the extractive approach.

For opinion summarization, there are a large number of reviews written by customers, and they are redundant in nature. Most summarization research generates phrases or short sentences that can convey information. Ganesan et al. [2] propose Opinosis, an unsupervised method using a graph structure created from words in the reviews, parts of speech, and locations of terms in the original sentences, to generate sentences according to a topic of interest, such as iPhone battery life. The graph is traversed to generate a summary whose grammar is checked against four predefined templates. Acceptable sentences are then scored and ranked based on term frequencies. Filippova [3] presents a multi-sentence compression method using a shortest path algorithm based on term frequencies. A simple grammatical checking process is included for English and Spanish. Micropinion [4] generates understandable short phrases of 2-7 words long by using a publicly available n-gram model and a depth first search to concatenate bigrams. Sentences are structurally examined. The score of a sentence is calculated from probabilities of term co-occurrences and readability.

Most opinion summarization works are proposed for Western languages. However, in Thai language words are written continuously without space, and many reviews are written informally. Identifying word boundary is shown to be a difficult and inaccurate task, thus accurate word identification, post tagging, and grammar checking cannot be relied on in a summarization method. There has not been previous research on abstractive summarization of Thai opinion texts. This research proposes a technique to solve that problem. Our technique is fully unsupervised, domain independent, and employs no grammar or linguistic annotation while using important text segments, redundancies of words, and writing structures in the review texts.

2. Proposed Technique

The proposed technique consists of 3 main processes: (1) segment extraction, (2) candidate phrase construction, and (3) summary generation. The segment extraction process selects important text segments that convey meanings of the reviews using a graph ranking algorithm. Bigrams from extracted segments with strong relationships between words are used to create a word graph which is traversed to construct a set of candidate phrases which are then scored. Similar phrases are grouped together, and the top-ranked phrase from each group is included in the summary.

2.1. Segment Extraction

The purpose of this step is to extract important segments that capture customer opinions from a set of reviews.

2.1.1. Thai Word Identification

In Thai language, there is no symbol to identify a word or the end of a sentence. Reviews are written in free forms. Though understandable by readers, they are generally neither complete sentences nor grammatically correct. In this research, text segments are character strings from customer reviews which are separated by special symbols (“?”, “.”, “:”,“;”, “*”,“-”, or whitespace), as defined in our previous work [1]. In addition, unlike in English word segmentation in Thai and in many other Asian languages is more complex because those languages do not have explicit word boundary delimiters [5]. In this work, word segmentation is performed by the longest matching algorithm in the SWATH software [6]. A shortlist of stop words is used which includes non-meaningful words, e.g., ค่ะ, ครับ, etc.

2.1.2. Word-Segment Matrix Compression

Each text segment S_j is represented as a vector $<f_{1j}, f_{2j}, \ldots, f_{mn}>$ where f_{ij} is the frequency of term i in segment j, m is the total number of terms in the entire reviews, and n is the number of segments in the reviews. A word-segment matrix A with n rows and m columns is created. The organization of the matrix assumes that all words are independent which may not generally be true in practice. A singular value decomposition (SVD) [7] is performed to compress the matrix into a lower dimensional feature space which can uncover hidden relationships among features and segments, and reduce effects of noises in segment characteristics. SVD decomposes matrix A into three components: an orthogonal

matrix of singular values, where $r = min(m, n)$, and the left and the right singular vectors (i.e., U and V, respectively). By keeping $k < r$ largest values of the singular matrix along with their corresponding columns in U and V, the resulting matrix is a matrix of rank k which is closest to the original matrix A in the least square sense. With respect to this new space of k dimensions, the words are no longer independent from one another.

2.1.3. Segment Extraction

A segment graph G is constructed from the compressed word-segment matrix. $G = (V, E)$ is a segment graph with a set of vertices V and a set of edges (or links) E where $V = \{S_1, S_2, ..., S_n\}$; S_i is the i-th segment in the reviews; and E is a subset of $V \times V$. A segment S_i is defined as a vector $<f_{1i}, f_{2i}, ..., f_{ki}>$ where f_{li} is the frequency of feature l for segment i, and k is the total number of features. Degree of similarity between segment S_i and segment S_j becomes the weight of edge between the nodes representing the two segments. Similarity between segments S_i and S_j can be calculated as follows:

$$similarity(S_i, S_j) = \frac{\sum_{v=1}^{k} f_{s_{iv}} * f_{s_{jv}}}{\sqrt{\sum_{v=1}^{k} f_{s_{iv}}^2} * \sqrt{\sum_{v=1}^{n} f_{s_{jv}}^2}}$$

A segment graph is an undirected weighted graph with edges placed between segment pairs with sufficient similarities. Each segment node will be assigned a significance score, using the Hopfield network algorithm [8]. The algorithm performs a parallel relaxation search in which nodes are activated in parallel, and activation values from different nodes are combined for each individual node. Neighboring nodes are traversed in order until the activation levels of nodes in the network converge. In the context of a segment graph, the graph can be viewed as a network whose nodes are represented by neurons, and edges are represented by synaptic links. The process terminates when there is no significant difference in terms of output between two consecutive iterations. The algorithm can be described as follows:

Initial State: The algorithm is initialized by

$$u_i(0) = 1,\ 0 \le i \le n\text{-}1$$

where $u_i(t)$ is the score of node i at iteration t.

Activation and Update State: Output of each node is calculated as follows:

$$u_i(t+1) = sigmoid[net_j],\ 0 \le j \le n\text{-}1$$

where $net_j = \sum_{n=0}^{n-1} w_{ij}\ u_i(t)$ is input passed through the activation function, and w_{ij} is the weight of synaptic link between S_i and S_j, and

$$sigmoid\ [net_j] = \frac{1}{1+\exp[\frac{\theta_j - net_j}{\theta_j}]}$$

where θ_j is a bias, and θ_o is an adjustable constant.

Stable State: Repeat the iteration until convergence. The stable state is achieved when sum of the error at every node in the network falls below a given threshold (ε).

$$\sum_{j=1}^{n-1} | u_j(t+1) - u_j(t) | \leq \varepsilon$$

Outputting State: After the network converges, the resulting outputs become the final significance scores of the corresponding segments.

Once the algorithm terminates, we have a score u_i for every segment i. The scores are sorted in a descending order. The top R segments are selected as the source of abstraction in further steps. The parameter R will be studied in the experiments.

2.2. Candidate Phrase Construction

From the set of selected segments, word-based bigrams are extracted and used to create a word graph. The graph is then traversed to generate a candidate phrase set. The graph is created from every bigram whose preceding word w_i has a strong collocation strength with the following word w_j. The collocation strength is calculated by Pointwise Mutual Information [9] biased toward bigrams that occur in many segments and thus represent legitimate word sequences. The collocation strength of a bigram (w_i,w_j) can be calculated as follows:

$$collocation(w_i, w_j) = freq(w_i, w_j) * \log_2\left(\frac{P(w_i, w_j)}{P(w_i) * P(w_j)}\right)$$

where $P(w_i,w_j)$ is the co-occurrence probability of the bigram, $P(w_i)$ and $P(w_j)$ are the probabilities of occurrences of w_i and w_j, respectively, and $freq(w_i,w_j)$ is the co-occurrence frequency of a word pair (w_i,w_j). A bigram with collocation strength greater than the collocation threshold is considered valid and used in the phrase scoring process. The collocation threshold can be calculated as follows:

$$collocation\ threshold = \log_2(m)$$

where m is the total number of unique words in the entire reviews. The result is a bigram matrix representation of a directed word graph, as shown in Figure 1.

wi/wj	อ่าน	อยาก	แล้ว	เป็น	ข้าราชการ	หนังสือ	ที่	ดี	มาก
อ่าน			1						
อยาก				1					
แล้ว		1							
เป็น					1	1			
ข้าราชการ									
หนังสือ							1		
ที่								1	
ดี									1
มาก									

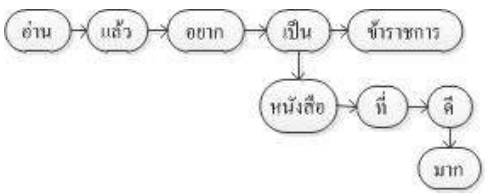

Figure 1. Bigram matrix and the corresponding word graph

In a word graph, a vertex represents a word in the matrix, and a directed edge connects a word in row w_i to another word in column w_j if the bigram (w_i,w_j) has a sufficiently strong collocation strength. A graph is traversed by a modified version of the depth first search (DFS) [10] where each preceding word w_i in bigrams takes turn to be the initial vertex, resulting in a set of candidate phrases. The word graph traversal algorithm is shown in Figure 2, and examples of phrases generated from the algorithm are shown in Figure 3. A phrase can be seen as a sequence of sub-phrases where a sub-phrase is a longest sequence of words that appear in any review.

Word Graph Traversal Algorithm
// Input: a word graph G
// Output: a set of candidate phrases S
$S = \{\}$
for $w_i \in$ preceding words in the word graph
    $S = S \cup$ {results of a depth first search on G starting at w_i}
end

Figure 2. Word graph traversal algorithm

เป็นหนังสือ**ดี**่มาก	อ่านแล้วอยากเป็นหนังสือ**ดี**่มาก
น่าอ่านแล้วอยากเป็นหนังสือ**ดี**่มาก	อยากเป็นหนังสือ**ดี**่มาก
แล้วอยากเป็นหนังสือ**ดี**่มาก	อ่านแล้วอยากเป็นข้าราชการ
หนังสือ**ดี**่มาก	น่าอ่านแล้วอยากเป็นข้าราชการ
ดี่มาก	แล้วอยากเป็นข้าราชการ
อยากเป็นข้าราชการ	ดีมาก
เป็นข้าราชการ	เล่มนี้

Figure 3. Examples of phrases resulted from the word graph traversal

Each generated phrase is then scored according to the following equation which takes into account words' importance represented by the Hopfield network algorithm scores of a word network, collocation strengths between consecutive words, lengths of sub-phrases, and is penalized by the mixture of sub-phrases from many reviews and the phrase length.

$$phrase\ score = \frac{\sum_{j=1}^{n}\left[\left\{\sum_{k=1}^{l-1}(collocation(w_k, w_{k+1}) * (Hopfield(w_k) + Hopfield(w_{k+1}))\right\} * \log_2(l_j)\right]}{l * n}$$

where n is the number of sub-phrases combining into the phrase, l_j is the length of sub-phrase j, and l is the length of phrase. The denominator captures actual writing structures in the review texts which promote understandability of the summary without using a grammar or linguistic annotation. Phrases with high scores are selected to generate a summary in the next section.

2.3. Abstractive Summary Generation.

In the final step, phrases with high scores are grouped according to the algorithm shown in Figure 4. Similarities between phrases are measured by the cosine similarity [11]. The phrase with the highest score in each group is included in the summary, to reduce duplicate information. Effects of the grouping threshold T will be studied in the experiments.

Phrase Grouping Algorithm
//Input: a list of high-scored phrases S, ordered by their scores in a descending order
//Output: a set of phrase groups

```
Assign the first phrase s1 as the representative for group 1.
for si ∈ in S
   calculate the similarity between si and the representative of each existing group sk.
   if (cosine similarity(sk,si) > threshold T )
        add the item to the corresponding group
        recalculate the group representative
   else
        use si to initiate a new group.
   end
   si = si+1
end
```

Figure 4. Phrase grouping algorithm

3. Experimental Evaluations

In this section, the proposed technique is evaluated and compared against two baseline methods, using actual customer reviews in Thai language.

3.1. Data Set

There is no standard test collection in Thai language, especially for opinion summarization. In this research, we gather data from the most popular online cosmetic website in Thailand www.jeban.com which contains reviews on a variety of products. Fifty products are picked randomly. For each product, reference summaries are created manually by 4 Thai female assessors with master degrees who are familiar with cosmetic products. The technique consists of two models: *Local Model* where the term collocation statistics are calculated only from the set of segments extracted by the segment extraction process, and *Global Model* where the term collocation statistics are calculated from the entire set of customer reviews.

3.2. Evaluation Metrics

In order to evaluate and compare techniques, two types of evaluations are performed: quantitative and qualitative evaluations. Quantitative evaluations intend to measure resemblance between generated summaries and reference (human) summaries. ROUGE [12] is popularly used as the main measure for text summarization problems. In our experiments, we use ROUGE-1, ROUGE-2 and ROUGE-SU4 measures. ROUGE-1 and ROUGE-2 have been shown to have most correlations with human summaries [2] while higher order ROUGE-N scores (N > 1) estimate the fluency of summaries.

For qualitative evaluations, three dimensions are measured: informative, grammatical, and non-redundancy aspects of a summary. The informative aspect measures how much users can learn from the summary, the grammatical aspect measures readability of the summary, and the non-redundancy aspect measures the uniqueness of phrases without unnecessary repetitions of facts in the summary. Each of these aspects is given a score from 1 (minimum) to 3 (maximum) by every human assessor. Two extractive baseline models are used to compare with the proposed technique which are: Baseline 1 where the text segments selected by the segment extraction process are grouped, and the phrase with the highest phrase score from each group is included in the summary; and Baseline 2 where the text segments selected by the segment extraction process are examined by human assessors, and only segments with non-redundant meanings are included in the summary.

3.3. Effects of Model Parameters

Two important parameters that can affect the performance of the proposed technique are the top R percent of segments to be extracted from the reviews and the grouping threshold T. We randomly sample 30% of the 50 products used in our study and vary the combinations of the two parameters, i.e., $R \in \{10\%, 20\%, \cdots, 90\%\}$ and $T \in \{0.1, 0.2, \cdots, 0.9\}$, and calculate the average F-measures over the sample. The results show that the combination of $R = 20\%$ and $T = 0.5$ yields the highest values of all

three ROUGE measures. Figure 5 shows the performance of the local model with T = 0.5 while the value of R is varied from 10% to 90%. The global model is not affected by R. We can see that the model performs the best when the top 20% of segments are extracted while at other values the model does not perform equally well.

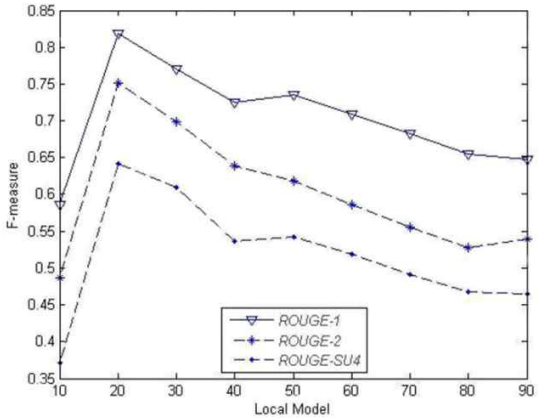

Figure 5. Effects of segment subset sizes on the local model (T is set at 0.5)

Figure 6 shows the effects of varying the values of the grouping threshold T from 0.1 to 0.9 when R is set at 20%. We can see that in all three ROUGE measures, the threshold of 0.5 yields the highest performance in general. Thus, in further experiments, we will set the values of parameters R and T at 20% and 0.5, respectively.

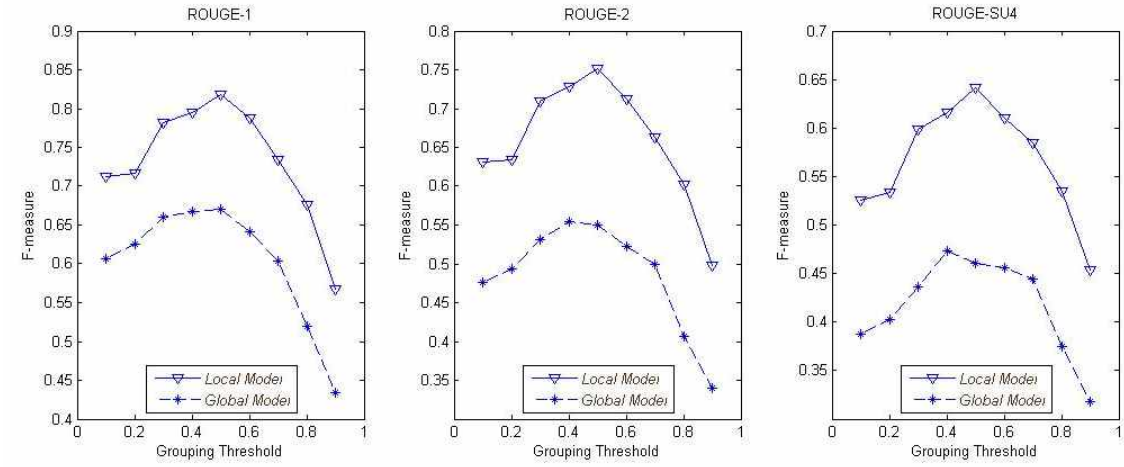

Figure 6. Effects of the grouping threshold T (R is set at 20%)

3.3.1. Quantitative Evaluations

We study the performance of the proposed technique (the local and global models) and compare it with the two baseline models quantitatively, using the three standard ROGUE measures. The results are shown in Table 1.

Table 1. Performance of the proposed and the baseline models

	Recall			Precision			F-Measure		
	Rouge-1	Rouge-2	Rouge-SU4	Rouge-1	Rouge-2	Rouge-SU4	Rouge-1	Rouge-2	Rouge-SU4
Local Model	0.6882	0.5620	0.4480	0.9014	0.9238	0.9474	0.7583	0.6708	0.5737
Global Model	0.5804	0.4262	0.3387	0.7945	0.8291	0.8661	0.6511	0.5395	0.4619
Baseline1	0.7321	0.4667	0.4557	0.5604	0.8099	0.7644	0.6089	0.5559	0.5332
Baseline2	0.7058	0.4884	0.4676	0.5132	0.6506	0.6246	0.5686	0.5181	0.4870

From Table 1, the results show that the local model is more effective than the global model in every measure. This shows the effectiveness of the segment extraction process where a set of representative segments is selected from the reviews. In terms of recall, the baseline models have higher recall than do our models in ROUGE-1 since baseline models returns actual segments from the review texts which contain a large number of single words, relative to the proposed model, however, it will hurt the precision as shown in the table. In other ROUGE measures which are based on bigrams and skip-bigrams, the local model yields the highest recalls. In terms of precision, the two proposed models are more effective than are the baselines, as discussed above. Overall, the local model performs the best quantitatively.

3.3.2. Qualitative Evaluations

The quantitative evaluations in the previous section do not show quality of summary in the eyes of the readers. Table 2 shows the qualitative results, averaged over all assessors. In the informative aspect, both proposed models provide more information to readers than do the baseline models which miss some points in the reviews. In the grammatical aspect, the local model is shown to generate more readable summary than does the global model. Since review texts are understandable by human, and the baseline models take actual phrases from the reviews, thus the grammatical aspect is not applicable to those models. In the non-redundancy aspect, Baseline 1 performs well since phrases are selected from the phrase grouping process which yields summaries with less redundancy than does Baseline 2. However, the local model is found to perform the best in this non-redundancy aspect. Overall, we can see that the local model produces the highest quality summary among the models and methods under study, across both quantitative and qualitative measures.

Table 2. Quality of the summaries generated by different models

	Baseline1	Baseline2	Local Model	Global Model
Informative	1.8235	1.7235	2.2353	2.1176
Grammatically	N/A	N/A	2.1176	2.0588
Non-redundancy	2.1176	1.8592	2.2235	2.1471

4. Conclusion

In this research, an automatic abstractive opinion summarization technique for reviews written in Thai language is presented. The technique begins with an extraction of important text segments representing opinions of customers using a graph ranking algorithm. Bigrams are generated from the segments, and those with strong collocation strengths are used to create a word graph. Through a traversal, a set of candidate phrases are created and ranked according to words' importance,

collocations among words, and structures of the review texts. Highly similar phrases are grouped, and the top-ranked phrase in each group is included in the summary. The technique is fully unsupervised, domain independent, and not relying on grammar or linguistic annotation while employing important text segments extracted, redundancies, and writing structures in the review texts. Two models which are parts of the proposed technique are studied: the local and the global models. Their major difference is in the bigram matrix construction which leads to different summary phrases to be generated. Both models are evaluated and compared with 2 extractive baseline methods quantitatively using 3 standard text summarization measures, i.e., ROUGE-1, ROUGE-2, and ROUGE-SU4, and qualitatively using 3 aspects which are informative, grammatical, and non-redundancy. The results show that the proposed local model generates the summaries most resembling human summaries as measured by ROUGEs and most quality as measured by the 3 qualitative measures.

5. References

[1] Ohm Sornil, K. Gree-ut, "An Automatic Text Summarization Approach using Content-Based and Graph-Based Characteristics", In 2006 IEEE Conference on Cybernetics and Intelligent Systems, pp. 1–6, 2006.

[2] Kavita Ganesan, ChengXiang Zhai, Jiawei Han, "Opinosis: A Graph-based Approach to Abstractive Summarization of Highly Redundant Opinions", In Proceedings of the 23rd International Conference on Computational Linguistics, pp. 340–348, 2010.

[3] Katja Filippova, "Multi-sentence Compression: Finding Shortest Paths in Word Graphs", In Proceedings of the 23rd International Conference on Computational Linguistics, pp. 322–330, 2010.

[4] Kavita Ganesan, ChengXiang Zhai, Evelyne Viegas, "Micropinion Generation: An Unsupervised Approach to Generating Ultra-concise Summaries of Opinions", In Proceedings of the 21st International Conference on World Wide Web, pp. 869–878, 2012.

[5] Poramin Bheganan, Richi Nayak, Yue Xu, "Thai Word Segmentation with Hidden Markov Model and Decision Tree", Advances in Knowledge Discovery and Data Mining, T. Theeramunkong, B. Kijsirikul, N. Cercone, and T.-B. Ho, Eds. Springer Berlin Heidelberg, pp. 74–85, 2009.

[6] Paisarn Charoenpornsawat, Virach Sornlertlamvanich, "Automatic Sentence Break Disambiguation for Thai", In Proceedings of International Conference on Computer Processing of Oriental Languages (ICCPOL), pp. 231-235. 2001.

[7] Kirk Baker, Singular Value Decomposition Tutorial, 2nd ed., The Ohio State University, USA, 2013.

[8] H. Chen, T. Ng, "An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): symbolic branch-and-bound search vs. connectionist Hopfield net activation", Journal of the American Society for Information Science, vol. 46, no. 5, pp. 348–369, 1995.

[9] Om P. Damani, "Improving Pointwise Mutual Information (PMI) by Incorporating Significant Co-occurrence", In Conference on Computational Natural Language Learning (CoNLL), 2013.

[10] Michael T. Goodrich, Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, John Wiley & Sons Inc.,USA, 2001.

[11] Anna Huang, "Similarity Measures for Text Document Clustering", In Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), pp. 49–56, vol. 6, 2008.

[12] Chin-yew Lin, "Rouge: a package for automatic evaluation of summaries", In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), pp. 25–26, 2004.