An Ensemble Similarity Model for Short Text Retrieval

Arifah Che Alhadi(✉), Aziz Deraman, Masita@Masila Abdul Jalil, Wan Nural Jawahir Wan Yussof, and Akashah Amin Mohamed

School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia
{arifah_hadi,a.d,masita,wannurwy}@umt.edu.my, akashahamin@gmail.com
http://www.umt.edu.my

Abstract. The rapid growth of World Wide Web has extended Information Retrieval related technology such as queries for information needs become more easily accessible. One such platform is online question answering (QA). Online community can posting questions and get direct response for their special information needs using various platforms. It creates large unorganized repositories of valuable knowledge resources. Effective QA retrieval is required to make these repositories accessible to fulfill users information requests quickly. The repositories might contained similar questions and answer to users newly asked question. This paper explores the similarity-based models for the QA system to rank search result candidates. We used Damerau-Levenshtein distance and cosine similarity model to obtain ranking scores between the question posted by the registered user and a similar candidate questions in repository. Empirical experimental results indicate that our proposed ensemble models are very encouraging and give a significantly better similarity value to improve search ranking results.

Keywords: Ensemble similarity model · Damerau-Levenshtein Distance · Cosine · Information retrieval

1 Introduction

The growth of online forum discussion or community-based question answering (CQA) sites is accompanied by a huge amount of potentially useful information that can be mined from their repositories. Online forum discussion is also used as a platform for distance learning where users can ask and discuss with the expert regarding their problems in education or working task. Online forums such as Stack Overflow, DreamInCode, MSDN, Tek-Tips and others have become popular platform used by computer users today.

Some users might also have the same problems and repeat a previously asked question without realizing it since there is no structure that organizes similar questions in an online forum discussion. All these sites are supported by experts whom are knowledgeable in providing answers on online forum. Question maybe

O. Gervasi et al. (Eds.): ICCSA 2017, Part I, LNCS 10404, pp. 20–29, 2017.
DOI: 10.1007/978-3-319-62392-4_2

left remain unanswered with many reasons like experts already answered the similar question and ask user to find themselves. This leads to the question starvation.

With a huge amount of questions, manual search is time consuming and impractical. Users turns to search engines to find the desired information. During this process, misspelled query terms is one of the main factors that affect the poor search result. Cucerzan et al., Duan et al. and Martins et al. mentioned that spelling errors frequently occurred in queries compared to written texts [6,8,18].

The widely used approach to identify similar text through lexical and semantic are based on string, corpus and knowledge [10]. There are also previous works that combines several techniques using ensemble or hybrid approach [3,16,19,20].

Even though ensemble or hybrid similarity model has shown significant contribution, it still suffers from the data-sparsity and noise in short texts such us the use of abbreviation, slang, misspelled, symbols and short form. This is due to the fact that current ensemble similarity model mostly rely on the third source (corpus or dictionary) and word-based model, hence insufficient and limited to deal with noise.

To overcome this issue, this research is proposed to analyze identified similarity model for short text retrieval by proposing an ensemble similarity model combining with edit distance model to handle misspell, as well as demonstrating the applicability of the proposed ensemble model.

In this paper, we work on measuring the similarity model between question and answer (QA) messages in online forum discussion using Damerau-Levenshtein distance (DLD) [7], and cosine similarity with *TF.IDF weights*. Our approach was based on works by [15,20] by using the same vector space model (VSM) and combined with edit distance similarity model.

2 Related Works

Extracting questions and answers from online forums are increasingly receiving academic attention. The major focus of previous research efforts on question answering is on Community-based Question Answering sites [5,22]. Cong et al. [5] used sequential pattern as feature to automatically distinguish between questions and non-questions in three different online forums (TripAdvisor, LonelyPlanet and BootsnAll). In addition, the classical features like 5W1H words and the presence of question marks were also used to detect the questions. They applied a graph based propagation method to identify and rank the candidate answers for the question in the same thread. Cosine similarity was used to measure the similarity between question answer pairs. The technique used was able to effectively extract question answer pairs and the average score for cosine similarity measured was 0.84.

Shtok et al. [22] reused the past resolved questions repository to answer the new question based on cosine similarity measure of question's titles. They calculate similarity score between two questions titles and bodies, entire title+body texts and entire text of each question and answer. However, based on our empirical analysis, using cosine similarity alone was not effective in finding similar

question. This is due to the high dependency of the cosine model on the availability of exact terms in both questions.

Questions retrieval methods based on machine translation models was proposed by Jeon et al. [12] to expand queries and generate translation probabilities between words in similar questions pair. The translation is used to measure the likelihood that a candidate question is matches to the query. Nevertheless, such translation models are less sufficient on short texts queries.

The dictionary or corpus is also used to enrich short text to increase the similarity value for compared sentences [9,16,20]. This model relies on additional information from other source. For current growth of short text media platform, user tend to compact the text by using abbreviation, slangs, jargon, symbol or short form created by themselves [1,16] and might also lead to misspelled words. Then the use of additional corpus decrease the effectiveness in identifying the semantic or similarity of texts because the occurrence of spelling errors will reduce the similarity value due to unable to detect and match the words in corpus.

Some researcher exploited the strengths and weakness of hybrid or ensemble techniques by combining the various similarity measures [3,16,19,20]. Hybrid model based on semantic word embeddings and tf-idf information is used by [3] to reduce the impact of less informative terms and mentioned that the combination leads to a better model for semantic content within very short texts. [19] measures similarity for short queries from Web search logs. The results showed that lexical matching and probabilistic methods are good in finding semantically identical matches and interesting topically related matches, respectively. It was shown that the combination of lexical, stemmed, and probabilistic matches results were better than any method alone.

Meanwhile Noah et al. [20] explored the potential of word order similarity and semantic similarity as an ensemble method to classify semantic Malay sentences. The calculation of word order similarities relied on vector similarity measures (cosine) however have been proven less effective based on our empirical analysis. The use of open dictionary shows the dependency of third party source.

Spelling correction for search queries also gained attention in previous works [6,8,14,18]. An automatic spelling correction will improve the quality of search result retrieval. The major types of spelling error are typographical errors (insertion, deletion, substitution, transposition), word boundaries (concatenation, splitting) or unfamiliar new words [8,14]. To overcome the spelling errors, Hidden Markov model [14], Damerau-Levenshtein distance [2], Levenshtein distance [13], Markov n-gram transformation model [8] were applied.

3 Proposed Ensemble Similarity Model

In this paper an ensemble similarity model is proposed to analyze the similarity of questions in QA archive. To the best of our knowledge, there is no previous research work using our proposed similarity methods to analysis the QA for online forum discussion. A general overview of the processes is shown in Fig. 1.

The proposed method is a linear combination of VSM and edit-distance model similarity. For the VSM, we consider the weight of word occurrence and for edit distance, we utilize the minimum number of editions required to transform one string to the other.

We simplify the process by skipping the stemming and stop word removal. This pre-processing method was applied to reduce noise of textual data. However work by Gao et al. [9], Saif et al. [21], Hu et al. [11] and Martínez-Cámara et al. [17] claimed that stemming and word removal process negatively impacts the performance of short text analysis. Saif et al. [21] also mentioned that two major limitation of existing stop word list is too generic and outdated due to the new information and terms are continuously increasing.

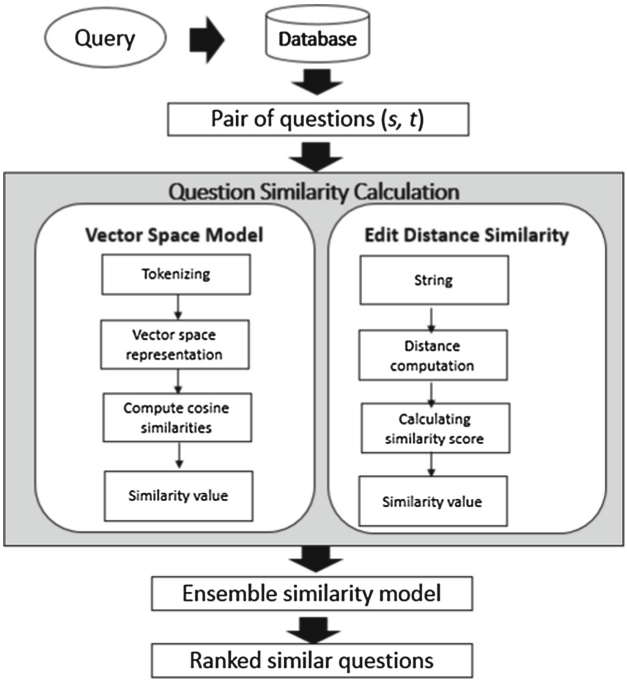


Fig. 1. Overview of the proposed model

A question in Stack Overflow, DreamInCode, MSDN and Yahoo! Answer contains title, question summary, and a body that have detailed explanation of the question. In most cases, when we refer to the question in this paper, it is actually referring to the question title. Jeon et al. [12] has demonstrated that using the question title for retrieval of similar questions is scored the highest in terms of effectiveness.

For each question, a similarity value is calculated for each similarity model. Using the VSM, question will be tokenized for breaking a question into words, phrases or symbols. Then the tokens will form a vector to represent each question and is organized in terms-document matrix. Weight of each word is then calculated based on the IDFs (inverse document frequencies). Finally the cosine similarity is computed to get the similarity value.

Whereas for edit distance model, string-string matrix will be computed for calculating the distance for each question pair. After receiving the similarity distance, we calculate the similarity score to get the similarity value. Finally the question similarity is derived by combining VSM and edit distance similarity value.

We will briefly describe the aforementioned processes of our proposed ensemble similarity questions detection in the following subsections.

3.1 Vector Space Model

Questions and queries are both vectors and expressed as t-dimensional vectors. Each term T, in a question or query j, is given a real valued weight w_{Tj}.

$$d_j = (w_{1j}, w_{2j}, \ldots, w_{Tj}) \tag{1}$$

Let m is a collection of documents that represent n term-document matrix. The weight (w) of a term in the document corresponds to an entry in the matrix. If the term does not exist in the document, then w is equal to zero.

The cosine similarity is measured by finding the cosine angle between two vectors of N dimensions. It is used to represent objects with different frequencies of its attributes. Documents are an example of objects that may have different frequencies of its attributes or words. Like Jaccard coefficient, cosine measure only considers attributes that present at least in one of the two objects being analyzed. In the documents example all the words would be the set of attributes, but for each document most of them would be zero valued. If the 0-0 matches were considered then documents in general would be highly similar. The cosine similarity is defined by:

$$cos(s,t) = \frac{s \cdot t}{\| s \| \| t \|} \tag{2}$$

The distance between the two vectors is an indication of the similarity of the two texts. The cosine of the angle between the two vectors is the most common distance measure. Assuming that we have a pair of questions, s and t of which:

- *s : Sipmle log java prgram*
- *t : Simple login java program*

where s is treated as the new query and t as the candidate question from the repository. The pairs representing spelling errors s paired with the correct spelling of the question t. Therefore, we will have a join set *st = [simple, sipmle, login, log, java, program, prgm]*. With the given of two vectors, we can measure

the similarity of s and t by calculating their cosine product based on the term-document matrix. Thus the similarity value for both questions is 0.25.

$$\begin{array}{c} \\ t \\ s \end{array} \begin{array}{c} \begin{array}{ccccccc} \textit{Simple} & \textit{sipmle} & \textit{login} & \textit{log} & \textit{java} & \textit{program} & \textit{prgm} \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array} \tag{3}$$

3.2 Edit-distance Similarity

DLD algorithm is used to calculate the similarity metrics which measuring the difference between two strings based on the number of changes that must be made to transform one strings to another resulting in a similarity or dissimilarity (distance) score [4]. Allowed changes are edition, insertion, deletion and transposition [7] - for example, the sequence (*siple*) can be converted to (*simple*) by one operations: transpose (*pm*).

Chen [4] also mentioned that similarity metric provides a floating point number indicating the level of similarity based on plain lexicographic match. Besides that, the DLD algorithm [7] also use to correct the spelling error as mentioned in [6,8,18], that spelling errors are most usually occurred in search queries.

Simply we can say that, the closeness of a match is measured in terms of the minimum number of operations necessary to convert the string into an exact match. We use the same question as mentioned earlier to calculate the DLD. For example, if the source string (in this case query input) is *sipmle* and the target string is *simple*, to transform *sipmle* to *simple*, we need to transpose *pm* to *simple*, thus, DLD will be 1. A small distance means the pairwise questions are very similar.

Figure 2 illustrates all the steps involved in calculating the DLD between *sipmle* and *simple*. To derive the DLD for s and t, the string s-string t matrix is constructed as shown in Fig. 2(b).

M_{ij} is a matrix of string s compared with string t. The recursion value of $M[i-1, j-1]$ is 0 if $s[i]$ and $t[j]$ are the same strings as shown in Fig. 2(c). Otherwise if $s[i] \neq t[j]$, the recursion value is computed using the following conditions:

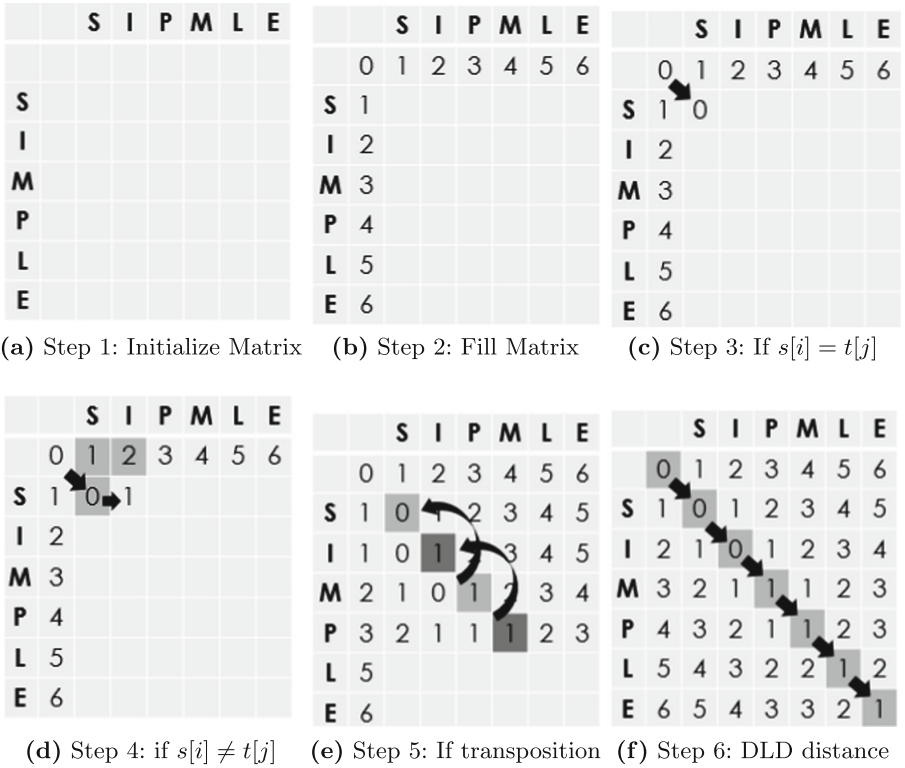
$$1 + min(M[i-1, j-1],\ M[i-1, j],\ M[i, j-1]) \tag{4}$$

For example, the value of cell M[1,2] is the minimum value of the cell $M[0,1]$, $M[0,2]$ and $M[2,1]$ plus 1 which given as follows:

$$M[1,2] = 1 + min(0, 1, 2) \tag{5}$$

The illustration is shown in Fig. 2(d). Figure 2(e) shows transposing the character p and m which result in the string *simple*. Transposition is allowed only between such characters that are adjacent already in the original string. So, in the case of transposition if $s[i] = t[j-1]$ and $s[i-1] = t[j]$, then the value is $1 + (M[i-2, j-2])$ where

$$\begin{aligned} M[3,3] &= 1 + m[3-2, 3-2] \\ &= 1 + m[1,1] \end{aligned} \tag{6}$$



(a) Step 1: Initialize Matrix **(b)** Step 2: Fill Matrix **(c)** Step 3: If $s[i] = t[j]$

(d) Step 4: if $s[i] \neq t[j]$ **(e)** Step 5: If transposition **(f)** Step 6: DLD distance

Fig. 2. Steps involve in calculating DLD

After retrieving the distance of DLD, the similarity score is then calculated by using the following formula:

$$Sim_{DamerauLevenshtein}(s,t) = 1 - \frac{DamerauLevenshteinDist(s,t)}{max(|s|,|t|)} \quad (7)$$

where *max* is the length of the longest of the two given texts (*s*,*t*) and DLD is the Damerau-Levenshtein Distance. Thus the similarity value of s and t is 0.84. A large value means the strings are very similar.

3.3 Ensemble Similarity Measure

The ensemble similarity measure will calculate the overall similarity between two questions by a linear combinations as follows:

$$Sim_{overall} = \delta Sim_{word} + (1 - \delta) Sim_{editDistance} \quad (8)$$

where δ is a constant value between 0 and 1, which decides the contribution of the involved similarity measure. We use 0.5 as constant value similar with [15,20]. Thus the ensemble similarity value of texts (*s*,*t*) is 0.55.

4 Preliminary Results and Discussion

We make several experiments to test the proposed methods to get the required results. The dataset used in this paper were collected from Stack Overflow with a few data with spelling error. Based upon the previous string-based similarity measures, we can derive ensemble similarity measures. Table 1 shows the results of cosine similarity, DLD and ensemble for pairwise analysis on new question or queries (s) with question in repository (t). We test them with different input queries to show the similarity value for both methods compared with our ensemble method.

If we look at the target query 1, *simple login java program*, for the third comparison, DLD (0.84) still gives higher value than cosine similarity measure (0.25) even there is a spelling error for the input question. The same results is also obtained for the fourth comparison which are DLD (0.92) and cosine (0.50). However, for the eighth comparison, the DLD still gives higher similarity percentage (0.16), compare with cosine which is 0. It shows that, DLD cannot work alone to rank the search result because based on our human interpretation, both questions are totally different in structure and meaning. To overcome this problem, we calculate the average between both methods to get similarity percentage and rank the search result.

Table 1. Pair of questions with its similarity value

Question	Questions compared	Cosine	DLD	Ensemble
Target query 1				
Simple login java program				
	1. Simple login java program	1	1	1
	2. Sipmle login java code	0.50	0.72	0.61
	3. Sipmle log java prgram	0.25	0.84	0.55
	4. Smple logon java program	0.50	0.92	0.71
	5. Simple login with java code	0.67	0.59	0.63
	6. Java GUI simple login back end mysql	0.57	0.42	0.50
	7. VERY simple user login system in Java	0.57	0.38	0.48
	8. Just any code	0	0.16	0.8
Target query 2				
Getting value from database				
	1. Getting value from database	1	1	1
	2. Getting specific value from Database	0.89	0.75	0.82
	3. Get value from detabase	0.50	0.81	0.66
	4. Get value from SQL database	0.67	0.70	0.69
	5. How get value from database	0.67	0.74	0.71
	6. Select one value from database	0.67	0.77	0.72
	7. Retrieve bit value from database	0.67	0.72	0.70
	8. Get value from data base	0.45	0.81	0.63

Then if the new question is similar to the existing question in the repositories, the results will be ranked according to the similarity measure. Thus, we can reduce the rate of unanswered questions by automatically giving the answers based on similar question answered in repositories.

5 Conclusion

In this paper, an ensemble similarity measure has been proposed in providing better search and retrieval of QA in forum discussion. The extracted QA pairs could be used to reduce unanswered question by giving the answer of new question which is similar to the past questions in repositories. The preliminary results also have shown the potential use of our ensemble similarity measure in overcome the weakness of both models by resolving the spelling error for the input query. Both techniques are useful to calculate the similarities of text but both does not consider any semantic in context. They literally calculate the similarities in physical words or letters without consider the meaning of words or the context of words in the question structure. However, the strength of both models also influence the achieving better similarity percentage. In future, human experts are needed to verify the results obtained by our proposed similarity method.

References

1. Anson, S., Watson, H., Wadhwa, K., Metz, K.: Analysing social media data for disaster preparedness: understanding the opportunities and barriers faced by humanitarian actors. Int. J. Disaster Risk Reduction **21**, 131–139 (2017)
2. Bard, G.V.: Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric. In: Proceedings of the Fifth Australasian Symposium on ACSW Frontiers, ACSW 2007, vol. 68, pp. 117–124. Australian Computer Society Inc., Darlinghurst, Australia (2007)
3. Boom, C.D., Canneyt, S.V., Bohez, S., Demeester, T., Dhoedt, B.: Learning semantic similarity for very short texts. CoRR abs/1512.00765 (2015)
4. Chen, H.: String Metric and Word Similarity applied to Information Retrieval. Master's thesis, School of Computing. University of Eastern Findland (2012)
5. Cong, G., Wang, L., Lin, C.Y., Song, Y.I., Sun, Y.: Finding question-answer pairs from online forums. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 467–474, SIGIR 2008 (2008)
6. Cucerzan, S., Brill, E.: Spelling correction as an iterative process that exploits the collective knowledge of web users. In: Proceedings of EMNLP **4**, 293–300 (2004)
7. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Commun. ACM **7**(3), 171–176 (1964)
8. Duan, H., Hsu, B.J.P.: Online spelling correction for query completion. In: Proceedings of the 20th International Conference on World Wide Web, pp. 117–126, WWW 2011, USA. ACM, New York (2011)
9. Gao, L., Zhou, S., Guan, J.: Effectively classifying short texts by structured sparse representation with dictionary filtering. Inf. Sci. **323**(C), 130–142 (2015)

10. Gomaa, W.H., Fahmy, A.A.: Article: a survey of text similarity approaches. Int. J. Comput. Appl. **68**(13), 13–18 (2013)
11. Hu, X., Tang, L., Tang, J., Liu, H.: Exploiting social relations for sentiment analysis in microblogging. In: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 537–546, WSDM 2013, USA. ACM, New York (2013)
12. Jeon, J., Croft, W.B., Lee, J.H.: Finding similar questions in large question and answer archives. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 84–90, CIKM 2005, NY, USA. ACM, New York (2005)
13. Lhoussain, A.S., Hicham, G., Abdellah, Y.: Adaptating the levenshtein distance to contextual spelling correction. Int. J. Comput. Sci. Appl. **12**(1), 127–133 (2015)
14. Li, Y., Duan, H., Zhai, C.: A generalized hidden Markov model with discriminative training for query spelling correction. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 611–620, SIGIR 2012, USA. ACM, New York (2012)
15. Li, Y., McLean, D., Bandar, Z.A., O'Shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng. **18**(8), 1138–1150 (2006)
16. Lochter, J.V., Zanetti, R.F., Reller, D., Almeida, T.A.: Short text opinion detection using ensemble of classifiers and semantic indexing. Expert Syst. Appl. **62**, 243–249 (2016)
17. Martínez-Cámara, E., Montejo-Ráez, A., Martín-Valdivia, M.T., Ureña López, L.A.: Sinai: machine learning and emotion of the crowd for sentiment analysis in microblogs. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), vol. 2, Proceedings of the Seventh International Workshop on Semantic Evaluation, pp. 402–407, SemEval 2013. Association for Computational Linguistics, Atlanta, Georgia, USA, June 2013
18. Martins, B., Silva, M.J.: Spelling correction for search engine queries. In: Vicedo, J.L., Martínez-Barco, P., Muńoz, R., Saiz Noeda, M. (eds.) EsTAL 2004. LNCS (LNAI), vol. 3230, pp. 372–383. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30228-5_33
19. Metzler, D., Dumais, S., Meek, C.: Similarity measures for short segments of text. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 16–27. Springer, Heidelberg (2007). doi:10.1007/978-3-540-71496-5_5
20. Noah, S.A., Amruddin, A.Y., Omar, N.: Semantic similarity measures for malay sentences. In: Goh, D.H.-L., Cao, T.H., Sølvberg, I.T., Rasmussen, E. (eds.) ICADL 2007. LNCS, vol. 4822, pp. 117–126. Springer, Heidelberg (2007). doi:10.1007/978-3-540-77094-7_19
21. Saif, H., Fernandez, M., He, Y., Alani, H.: On stopwords, filtering and data sparsity for sentiment analysis of twitter. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014). European Language Resources Association (ELRA), Reykjavik, Iceland, May 2014
22. Shtok, A., Dror, G., Maarek, Y., Szpektor, I.: Learning from the past: answering new questions with past answers. In: Proceedings of the 21st International Conference on World Wide Web, pp. 759–768, WWW 2012 (2012)

http://www.springer.com/978-3-319-62391-7

Computational Science and Its Applications – ICCSA 2017
17th International Conference, Trieste, Italy, July 3-6, 2017, Proceedings, Part I
Gervasi, O.; Murgante, B.; Misra, S.; Borruso, G.; Torre, C.M.; Rocha, A.M.A.C.; Taniar, D.; Apduhan, B.O.; Stankova, E.; Cuzzocrea, A. (Eds.)
2017, XXXVI, 779 p. 293 illus., Softcover
ISBN: 978-3-319-62391-7