Re-examining Google Tri-grams Measure (GTM) Sentence Similarity

Wong Lin Juan Linda, Chih How Bong, and Nung Kion Lee

***Abstract*—This paper examines text similarity approach based on Google n-gram dataset. Google Tri-grams Measure (GTM) is an unsupervised text similarity measure. The paper investigates the sentence similarity of GTM which in turn reveals the approach's pitfalls. We also compared GTM's sentence similarity measures on Li-30 sentence pairs, Microsoft Research Paraphrase Corpus paraphrase, Kaggle Quora Question Pairs competition's dataset respectively against human judgement. Other sentence similarity measures are compared against GTM. We discovered GTM sentence similarity has a lot of weight on overlapped words count. However, despite the weakness, it still outperformed other replicated sentence similarity measures.**

***Index Terms*—Google trigrams, pitfalls, sentence similarity, text similarity, trigrams, unsupervised, word similarity.**

I. Introduction

Text are words and phrases. The two measures commonly used to gauge if two given text are similar are text similarity and text relatedness. Text similarity quantifies closeness of two texts, on the other hand, text relatedness is the degree of how two texts relate to each other. Theoretically, text relatedness is a function of word relatedness. Text relatedness measures are methods to quantify the relatedness of two texts while text similarity measures are methods that are used to identify how similar the texts to each other. According to Mihalcea [1], there is an obvious relatedness between two phrases like "We own a pet" and "I love animals", even though they are obviously dissimilar. Text similarity and relatedness are two of the important area in the field of natural language processing and they are widely applied in real life like, detecting plagiarism [2], automatic question answering [3] that return candidate answers by evaluating textual data and information retrieval [4] as in searching for related articles based on the keywords like Google and Yahoo search engines.

To date, text similarity is computed by using word and phrase similarity. TrWP [5] is an unsupervised text similarity approach using both word and phrase similarity. It is a Bag-of-Word-and-Phrase (BoWP) approach where phrase-pair (unigram vs bi-gram or bi-gram vs bi-gram) are used to computes the text similarity. It adopts Sum-Ratio (product of sum and ratio between minimum and maximum of two numbers) to capture the strength of association between two overlapping Google n-grams based on the statistics in the Google n-gram dataset of overlapping n-grams associated with the two compared texts [5].

There is no lack of literatures since researchers like Landauer [6], Mihalcea [7], Li *et al.* [8], and Lin [9] wo have produced various text similarity measures. Well-known works like LSA [6] uses Singular Value Decomposition (SVD) to analyze the statistical relationships among words to find the semantic representation of words in a reduced dimensional space. To derive similarity, corresponding word vectors are computed of its cosine angle to indicate text similarity. On the other hand, Li *et al.* [8] proposed a method that computes text similarity based on corpus statistics and syntactic information. The approach has also considered sequence of words of a text as it carries useful information and specific meaning. Liu [10] proposed a novel approach to compute short text similarity by considering semantic information, word order and the contribution of different parts of speech in a sentence. The overall sentence similarity is derived from a weighted combination of the distance between sub sequences.

In 2012, Islam [11] has reported that their proposed text similarity--Google Tri-grams Measure (GTM)--has outperformed many well-performed text similarities. Hence in this paper, we intend to detail how GTM works and at the same time, to highlight the pitfalls of the measures. Lastly, we will present some evidences to verify the pitfalls.

II. Google Tri-grams Measure (GTM) [11]

GTM is a distributional method that uses a Google n-gram dataset to find the inherent properties of similarity between texts. In general, GTM has two main components: word similarity and text similarity. The word similarity component is to derive word-word similarity which is the fundamental component that is required to derive the sentence similarity. The word-word scores are aggregated to deliver a score to represent text similarity.

A. Determining Word Similarity

The word similarity in GTM is derived through Google n-grams's tri-grams dataset. It takes into consideration all the tri-grams that begins and ends with the given pair of words regardless of their order. In additional, the most frequent unigram of each word is used to normalize the mean frequency of the tri-grams. The algorithm of the word similarity is described in detail in the following.

Given two words, w_a and w_b,

Manuscript received September 12, 2017; revised November 10, 2017.

The authors are with Faculty Computer Science and Information Technology, Universiti Malaysia Sarawak, Malaysia (e-mail: 15020282@siswa.unimas.my, chbong@unimas.my, nklee@unimas.my).

Step 1: First, obtain the highest unigram frequency from Google unigram dataset, which is represented as F_{max}.

Step 2: Obtain the frequency of unigram w_a as $f(w_a)$, and w_b as $f(w_b)$ from Google unigram dataset.

Step 3: Between the unigram frequency of w_a and unigram frequency of w_b, choose the frequency of the unigrams with minimum frequency as $min(f(w_a),f(w_b))$.

Step 4: Obtain the sum of the frequency of tri-grams that begins with w_a, ends with w_b as $f(w_a w_i w_b)$,

Step 5: Obtain the sum of the frequency of tri-grams that begins with w_b, end with w_a as $f(w_b w_i w_a)$.

Step 6: The information obtained from step 1 to step 5 is used to compute the word similarity which is defined as:

$$Sim\,(w_a, w_b) = \frac{\log \dfrac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times F_{\max}{}^{2}}{f(w_a) \times f(w_b) \times \min(f(w_a), f(w_b))}}{-2 \times \log \dfrac{\min(f(w_a), f(w_b))}{F_{\max}}} \tag{1}$$

In order to make sure that the $Sim(w_a,w_b)$ is always a positive number, there are three conditions as shown below.

$$Sim\,(w_a, w_b) = \begin{cases} \dfrac{\log \dfrac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times F_{\max}{}^{2}}{f(w_a) \times f(w_b) \times \min(f(w_a), f(w_b))}}{-2 \times \log \dfrac{\min(f(w_a), f(w_b))}{F_{\max}}} & if \quad \dfrac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times F_{\max}{}^{2}}{f(w_a) \times f(w_b) \times \min(f(w_a), f(w_b))} \geq 1 \\ \dfrac{\log 1.01}{-2 \times \log \dfrac{\min(f(w_a), f(w_b))}{F_{\max}}} & if \quad \dfrac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times F_{\max}{}^{2}}{f(w_a) \times f(w_b) \times \min(f(w_a), f(w_b))} < 1 \\ 0 & if \quad \dfrac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times F_{\max}{}^{2}}{f(w_a) \times f(w_b) \times \min(f(w_a), f(w_b))} = 0 \end{cases} \tag{2}$$

The word similarity is computed based on the equation 2 referring to the condition, $\frac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times F_{\max}{}^{2}}{f(w_a) \times f(w_b) \times \min(f(w_a), f(w_b))}$ which is calculated from the information collected from step 1 to step 5.

B. Determining Sentence Similarity

The main idea of this sentence similarity is to extract overlapped word from the sentence and construct a m by n matrix where m-rows and n-column consists of non-overlapped tokens from first text and second text respectively. Placing the texts in m-rows or n-column is decided by the length of texts which will be explained below. Each element in the matrix representing words similarity scores using the word similarity explain in Section 2.1. We then compute from the matrix the summation of mean and standard deviation of each row to derive an aggregated score to represent the sentence similarity score.

The sentence similarity measure consists of 7 steps as described below. Let T_1 and T_2 be the two input Texts.

Step 1: Pre-process the two inputs by removing special characters, punctuations and stop words in the texts.

Step 2: Calculate the number of tokens left after step 1. Let m and n be the number of tokens left of the two texts T_1 and T_2 respectively. Assuming that $T_1=\{a_1,a_2,a_3,\ldots,a_m\}$ and $T_2=\{b_1,b_2,b_3,\ldots,b_n\}$.

Step 3: Examine if $n \geqslant m$. If $m \geqslant n$, switch the order of the two input texts T_1 and T_2 to make sure that number of token in T_2, n is always greater than number of tokens in T_1, m.

Step 4: Compare each of the tokens of texts T_1 and T_2. Remove the overlapped tokens from T_1 and T_2. δ is the number of the overlapping words from both $T_1$and T_2. Therefore, T_1left with (m-δ) tokens and T_2 with (n-δ) tokens.

Step 5: Construct a semantic matrix M of $(m\text{-}\delta)\times(n\text{-}\delta)$ Where (m-δ) refers to row and (n-δ) refers to column of matrix

$$M = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{1j} & \cdots & \alpha_{1(n-\delta)} \\ \alpha_{21} & \alpha_{22} & \alpha_{2j} & \cdots & \alpha_{2(n-\delta)} \\ \alpha_{i1} & \alpha_{i2} & \alpha_{ij} & \ddots & \alpha_{i(n-\delta)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{(m-\delta)1} & \alpha_{(m-\delta)2} & \alpha_{(m-)1} & \cdots & \alpha_{(m-\delta)(n-\delta)} \end{bmatrix}$$

The elements of matrix are word-word similarity scores as defined in Section 2.1. For example, α_{ij} refers to row i and column j position in the matrix

Step 6: Compute mean (μ) and standard deviation (σ) of each row in matrix M. The equation of mean and standard deviation is as stated below.

$$\mu(\{a_1,\ldots,a_x\}) = \frac{1}{x}\sum_{i=1}^{x} a_i \tag{3}$$

$$\sigma(\{a_1,\ldots,a_x\}) = \sqrt{\frac{1}{x}\sum_{i=1}^{x}\left(a_i - \mu(\{a_1,\ldots,a_x\})\right)^2} \tag{4}$$

Calculate the summation of mean (μ) and standard deviation (σ) of each row in M. For each element in each row (m-δ), build a set of elements for each row A_i, $\{g_1,g_2,\ldots,g_x\}$, such that each element in A_i are those elements from matrix M that having score that is greater than the summation of mean and standard deviation (μ+σ) of each row. The mean of each row is $\mu(A_i)$ and these means of (m-δ) rows are added up to be $\sum_{i=1}^{m-\delta}\mu(A_i)$.

Step 7: Finally, the sentence similarity score is computed with the equation below.

$$S(T_1,T_2) = \frac{\left(\delta + \sum_{i=1}^{m-\sigma}\mu(A_i)\right)\times(m+n)}{2mn} \tag{5}$$

The sentence similarity score is scaled by reciprocal harmonic mean of m and n and the score is ranged between 0 and 1.

III. A Walkthrough Example of Calculating Sentence Similarity Score Using GTM

In the following, we take the sentence examples from Li *et al* [8] to illustrated the steps to compute the similarity score with GTM. Give two sentences,

T_1 : RAM keeps things being worked with, and

T_2 : The CPU uses RAM as a short-term memory storage.

Step 1: Pre-process the texts above by removing the stop-words, special characters and punctuation. This results in T_1= {RAM, keeps, things, worked} and T_2 = {CPU, uses, RAM, short, term, memory, storage}.

Step 2: T_1 contains 4 tokens while T_2 contains 7 tokens. Therefore, m = 4 and n = 7.

Step 3: Since number of tokens in T_2 is larger than T_1, there is no need to switch the order of the texts in the matrix.

Step 4: The two texts contains an overlapped word which is "RAM". Therefore, it will be removed from the two texts and the overlapped word score will be δ=1. After the removal of all overlapped word, the texts becoming T'_1 = {keeps, things, worked} and T'_2 = {CPU, uses, short, term, memory, storage}.

Step 5: Compute the word similarity score of each word in T'_1 and T'_2 as described in Section 2.1.

Step 6: Construct a matrix M of $(m\text{-}\delta)\times(n\text{-}\delta)$ where each element of the matrix corresponding to similarity score of the words.

	keeps	things	worked
CPU	0.617	0.508	0.000
uses	0.465	0.451	0.316
short	0.486	0.448	0.460
term	0.418	0.357	0.383
memory	0.558	0.474	0.407
storage	0.600	0.469	0.264

Fig. 1. Similarity score of pairwise word for sentence pair.

The above figure shows similarity score of the pairwise word.

Step 7: Calculate the summation of mean and standard deviation of each row.

	keeps	things	worked	µ+σ
CPU	0.617	0.508	0.000	0.644
uses	0.465	0.451	0.316	0.478
short	0.486	0.448	0.460	0.481
term	0.418	0.357	0.383	0.411
memory	0.558	0.474	0.407	0.541
storage	0.600	0.469	0.264	0.583

Fig. 2. Similarity score of pairwise word for sentence pair with sum of mean and standard deviation.

From the figure above, last column shows the summation of mean and standard deviation (μ+σ) of each row. The score of each element that is greater than (μ+σ) is used to construct sets of elements A= {{0}, {0}, {0.486}, {0.418}, {0.558}, {0.6}}. Therefore, $\sum_{i=1}^{m-\delta}\mu(A_i)=0$ is 2.062.

Step 8: Compute the sentence similarity score. Here, δ=1, $\sum_{i=1}^{m-\delta}\mu(A_i)=0$ =2.062, m=4 and n = 7.

Using equation 3, the sentence similarity of the two texts, SS(T_1 $T_{2,}$) = $\frac{(1+2.062)\times(4+7)}{2(4)(7)}$ =0.251.

IV. Pitfall in Deriving the Sentence Similarity Score

By examining Equation 3, we can learn that the similarity score composed of computing of overlapped words of two compared texts, number of tokens of both pre-processed texts, mean and standard derivation of the word similarity matrix, *M*.

From the walkthrough above, we also notice that the sentence similarity score is composed literally from the count of overlapped words. For example, for "*RAM keeps things being worked with*" and "*The CPU uses RAM as short-term memory*", the main components are 1 overlapped word, word similarity matrix mean is 2.062, and there were 5 tokens from the first and 7 tokens from the second text. Hence, the sentence similarity score could be derived as

$$\frac{\left(\delta+\sum_{i=1}^{m-\sigma}\mu(A_i)\right)\times(m+n)}{2mn}=\frac{(1+2.062)\times(4+7)}{2(4)(7)}=0.251.$$

However, for the sentence pair, "A gem is a jewel or stone that is used in jewellery." and "A jewel is a precious stone used to decorate valuable things that you wear such as rings or necklaces.". This pair of sentence is high semantically similar as the human annotated score in Li-30 sentence pair is 0.65. But GTM scored them with 0.45 due to the reason that the overlapped word count is low out of the total number of tokens in each text.

V. The Efficacy of GTM on Measuring Sentence Similarity

In this section, we report efficacy of GTM sentence similarity against three datasets which are Li30 [8] Microsoft Research Paraphrase Corpus (MSRPC) [12], and Kaggle Quora [13].

A. Li30[8]

Li30 dataset composed of 30 chosen sentences pairs subset from Rubenstein & Goodenough 65 sentence pairs [14]. The constructed dataset, has a similar procedure as Miller and Charles [15] which involves human participation to rate the similarity ranged between 1 to 4. The Pearson's correlation coefficient of different sentence similarity measure of the human annotated score is compared against sentence similarity measure by Li *et al*. [8] and LSA [6]. The results were recorded in Table III.

TABLE I: Results of Sentence Similarity Measures Against Li-30 Human Annotation

Sentence Similarity Measures	Correlation (r)
GTM [11]	0.853
Li *et al.* [8]	0.829
LSA [6]	0.646

From the results, we observed that GTM sentence similarity has the highest correlation score 0.853 which is slightly better than Li *et al.* [8] which is 0.829 while LSA[6] has the lowest score which is 0.646. Table IV details the similarity scores of each sentence pair with word pairs to represent each sentence pair respectively.

TABLE II: RESULTS OF GTM SENTENCE SIMILARITY AGAINST HUMAN ANNOTATION FROM LI-30'S DATASET

No.	Sentence Pairs	Human Annotation	GTM Sentence Similarity	Difference
1	Cord-Smile	0.01	0.00	0.01
5	Autograph-Shore	0.01	0.00	0.01
9	Asylum-Fruit	0.01	0.00	0.01
13	Boy-Rooster	0.11	0.00	0.11
17	Coast-Forest	0.13	0.17	0.04
21	Boy-Sage	0.04	0.00	0.04
25	Forest-Graveyard	0.07	0.13	0.06
29	Bird-Woodland	0.01	0.00	0.01
33	Hill-Woodland	0.15	0.21	0.06
37	Magician-Oracle	0.13	0.00	0.13
41	Oracle -Stage	0.28	0.00	0.28
47	Furnace-Stove	0.35	0.00	0.35
48	Magician-Wizard	0.36	0.15	0.21
49	Hill-Mound	0.29	0.00	0.29
50	Cord-String	0.47	0.17	0.30
51	Glass-Tumbler	0.14	0.16	0.02
52	Grin-Smile	0.49	0.25	0.24
53	Serf-Slave	0.48	0.32	0.16
54	Journey-Voyage	0.36	0.18	0.18
55	Autograph-Signature	0.41	0.19	0.22
56	Coast-Shore	0.59	0.31	0.28
57	Forest-Woodland	0.63	0.20	0.43
58	Implement-Tool	0.59	0.51	0.08
59	Cock-Rooster	0.86	0.75	0.11
60	Boy-Lad	0.58	0.50	0.08
61	Cushion-Pillow	0.52	0.13	0.39
62	Cemetery-Graveyard	0.77	0.38	0.39
63	Automobile-Car	0.56	0.32	0.24
64	Midday-Noon	0.96	0.75	0.21
65	Gem-Jewel	0.65	0.45	0.20

From the table above, we can observe that GTM sentence similarity's result is quite close to human annotated score. The average difference is 0.16.

Sentence pair "*Forest-Woodland*" has the greatest difference which is 0.43 from human annotated score. Word "*Forest*" represents sentence "*Woodland is land with a lot of trees*" while "*Woodland*" represents sentence "*A forest is a large area where trees grow close together*". The two sentences are semantically similar due to the reason that the sentences have low count on overlapped words, GTM sentence similarity returned a low score which is 0.20.

B. MSRPC[12]

We also report the efficacy of GTM against Li *et al* [8], Mihalcea [7], and LSA [6] using Microsoft Research Paraphrase Corpus (MSRPC) [12]. MSRPC composes of 5801 paraphrase sentence pairs that were collected from Web news. Among the 5801 sentence pairs, 4076 are regarded as training pair whereas 1725 are testing pairs. The sentence pairs are annotated by human judges with binary classification where 1 refers to paraphrase or semantic equivalence whereas 0 refers to non-paraphrase. The similarity measures are evaluated by F-1 score (harmonic mean of precision and recall) against human judgements.

TABLE III: RESULTS OF WORD SIMILARITY MEASURES AGAINST HUMAN ANNOTATION WITH MSR PARAPHRASE TEST DATASET

Sentence Similarity Measures	Precision (%)	Recall (%)	F-measure (%)
Li et. al [8]	44.21	66.49	53.11
Mihalcea [7]	66.49	66.49	66.49
Lsa [6]	62.68	66.67	55.10
GTM [11]	67.94	69.74	65.35

The results above show that GTM sentence similarity measure outperformed the other measures by obtaining highest accuracy precision and recall.

TABLE IV: EXAMPLES OF SENTENCE PAIRS WITH THE RESULTS

No	Sentence Pairs	Annotated Result	GTM Score	Similar? (Threshold= 0.492)
1	Scientists believed Stardust trapped thousands of particles of dust.	0	0.536	1
	Stardust was designed to gather thousands of dust particles streaming from Wild 2.			
2	West Nile Virus -- which is spread through infected mosquitoes -- is potentially fatal.	0	0.670	1
	West Nile is a bird virus that is spread to people by mosquitoes.			
3	The House barely had the necessary 100 members present for a quorum.	0	0.550	1
	A hundred House members are needed for a quorum.			
4	The Senate version has no coverage for annual costs between $4,450 and $5,800.	1	0.225	0
	They would receive no help with costs between $4,500 and $5,800.			
5	It estimated on Thursday it has a 51 percent market share in Europe.	1	0.417	0
	Boston Scientific said it has gained 51 percent of the coated-stent market in Europe.			
6	The launch marks the start of a new golden age in Mars exploration.	1	0.375	0
	The launch marks the start of a race to find life on another planet.			

In the following, we listed a few sentence pairs with annotated results which are extracted from MSR Paraphrase dataset [12]. The system output results obtained from the investigated method are being converted based on optimal threshold that is obtained from the trained results, which is 0.492. The threshold is identified by from the highest ratio of true positives to false positives. The converted output result is either 0 or 1 which represents similar or dissimilar.

From the table above, we can infer that non-similar texts are judged as similar because overlapped words contribute higher scores to the sentence similarity.

The texts "Scientists believed Stardust trapped thousands of particles of dust?" and "Stardust was designed to gather thousands of dust particles streaming from Wild 2." is annotated by human experts as not similar. However, the GTM sentence similarity method regarded them as similar in this example, the number of tokens in both sentence contains 4 overlapped words (stardust, thousands, particles, dust) out of total number of token 7 and number of token 8 from first sentence and second sentence respectively excluding stop words. The ratios of overlapped words are high in both sentence.

On the other hand, the pair "The launch marks the start of a new golden age in Mars exploration." and "The launch marks the start of a race to find life on another planet." (see No. 6 in Table VI) is annotated as similar by human judges, whereas GTM regarded them as not dissimilar. This is because after stop words removal, the ratio of overlapped word (launch, marks, start) to total tokens in both texts is 3:8. With that, the zero scores in the semantic matrix yield low sum of mean and standard deviation which directly lead to low GTM score. We also notice that overlapped words play a crucial factor to determine the similarity.

C. Kaggle Quora Question[13]

Besides using Microsoft Research Paraphrase Corpus (MSRPC), we also used dataset from Kaggle Quora Question Pairs Competition [13] which aim to predict which of the provided question pairs with the same meaning. Multiple datasets are used for the comparison is to prove there's no bias and provided with evidences. The dataset has been annotated by human experts. Same sentence similarity measures and GTM [11] sentence similarity is compared with the dataset and the result is shown in the table below.

TABLE V: RESULTS OF WORD SIMILARITY MEASURES AGAINST HUMAN ANNOTATION WITH KAGGLE QUORA QUESTION PAIRS COMPETITION'S DATASET

Sentence Similarity Measures	Precision (%)	Recall (%)	F-measure (%)
Li et. al [8]	62.07	61.19	65.25
Mihalcea [7]	65.96	63.63	68.61
Lsa [6]	53.61	57.34	62.99
GTM [11]	66.81	65.73	66.51

From this table, if we would compare with the results of MSRPC dataset from Table V, we can see GTM [11] is still recorded as highest precision and recall among the sentence similarity measures while Mihalcea [7] is still recorded as sentence similarity with the highest F-measure. For both datasets, same methods are used to make sure no bias in the comparison. The sentence pairs are compared with respective sentence similarity and threshold is obtained for each sentence similarity. The sentence similarity score of each pair of sentence is compared to the threshold. If the score is greater or equal to the threshold, the sentence similarity score of the pair of sentence will be converted to 1 and vice versa.

We listed a few sentence pairs from Kaggle Quora Question Pairs competition's dataset to show as evidence of pitfall of GTM sentence similarity measure which is priority on overlapped words in table below.

TABLE VI: EXAMPLES OF SENTENCE PAIRS WITH THE RESULTS

No	Sentence Pairs	Annotated Result	GTM Score	Similar? (Threshold = 0.535)
1	Why do we cry when we are happy and when we are sad?	0	0.667	1
	Why do we cry			
2	Is it normal for older men to be attracted to young women?	0	0.750	1
	Why am I attracted to older men?			
3	How can I stop being afraid of working?	0	0.667	1
	How do you stop being afraid of everything?			
4	What are some special cares for someone with a nose that gets stuffy during the night?	1	0.514	0
	How can I keep my nose from getting stuffy at night?			
5	How do we prepare for UPSC?	1	0.417	0
	How do I prepare for civil service?			
6	What causes a nightmare?	1	0.375	0
	What causes nightmares that seem real?			

The texts "*Why do we cry when we are happy and when we are sad?*" and "*Why do we cry*." is annotated by human experts as not similar. However, GTM sentence similarity method rendered them as similar. In this example, the number of tokens in both sentence contains 4 overlapped words (*why*, *do*, *we*, *cry*) out of total number of token 13 and number of token 4 from first sentence and second sentence respectively excluding stop words. The ratios of overlapped words are high in both sentence. Therefore, GTM sentence similarity return the score as 1, which is similar.

On the other hand, the 4th sentence pair from the table above, "What are some special cares for someone with a nose that gets stuffy during the night?" and "How can I keep my nose from getting stuffy at night?" is annotated as similar by human experts. However, the GTM sentence similarity score is 0, which is not similar. We examined the sentence pair, we can find only 3 overlapped words, (nose, stuffy, night) out of total number of tokens 14 and 8 from first and second sentence respectively excluding stop words. The ratios of overlapped words are low in both sentence pairs, thus dissimilar.

VI. CONCLUSION AND FUTURE WORK

In this paper, we examine and discuss the pitfalls of

GTM. Islam [3] reported GTM outperformed other sentence similarity measures. After evaluation, we discovered GTM scores the highest accuracy, precision and recall among the other replicated sentence similarity measures.

GTM has priority on overlapped words and these words significantly render the sentence similarity scores. Therefore, sentences that are not semantically similar but contains higher count on overlapped words has a high similarity score and vice versa. Despite the weakness, GTM still recorded higher performance than other sentence similarity measures as shown in section 5.

In the future, we would like to examine the efficiency and performance of GTM sentence similarity measures by replacing the GTM word similarity to other word similarity measures. This is to examine if GTM word similarity, which is state-of-the-art of GTM that uses trigrams data to compute the word similarity perform better than other word similarities in GTM sentence similarity measure.

Acknowledgment

I would like to thank Universiti Malaysia Sarawak (UNIMAS) and KPM who funded this research study through the grant FRGS/ICT07(04),1203,2014(04).

References

[1] G. Ignatow and R. Mihalcea, "Text mining: A guidebook for the social sciences," *Sage Publications,* 2016.

[2] M. Wise, "YAP3: Improved detection of similarities in computer program and other texts," *ACM SIGCSE Bulletin,* vol. 28, no. 1 pp. 130-134, 1996.

[3] G. Liu, Z. Lu, T. Hao, and W. Liu, "Automatic short text annotation for question answering system," in *Proc. International Conference on Web Information Systems and Technologies. Springer, Berlin, Heidelberg,* 2010.

[4] G. Salton, J. Allan, and C. Buckley, "Approaches to passage retrieval in full text information systems," in *Proc. of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[5] M. Rakib, A. Islam, and E. Milios, "TrWP: Text relatedness using word and phrase relatedness," *SemEval@ NAACL-HLT,* pp. 90-95, 2015.

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science,* vol. 41, no. 6, pp. 391-407, 1990.

[7] R. Mihalcea, C. Corley, and C. Strapparava. "Corpus-based and knowledge-based measures of text semantic similarity," in *Proc. of AAAI'06,* 2006.

[8] Y. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources, *IEEE Transactions on Knowledge and Data Engineering,* vol. 15, no. 4, pp. 871-882, 2003.

[9] D. Lin, "An information-theoretic definition of similarity," in *Proc. of the 15th ICML,* 1998, pp. 296-304.

[10] X. Liu, Y. Zhou, and R. Zheng, "Sentence similarity based on dynamic time warping," in *Proc. of International Conference on Semantic Computing,* 2007, pp. 250-256.

[11] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Trans. Knowl. Discov. from Data,* vol. 2, no. 2, 2008.

[12] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *Proc. of the 20th international conference on Computational Linguistics,* Moristown, NJ, USA: Association for Computational Linguistic, 2004.

[13] K. Csernal, "First quoradataset release: Question pairs," [Online]. Available: https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

[14] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Commun, ACM,* vol. 8, no. 10, pp. 627-633, 1965.

[15] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Language and Cognitive Processes,* vol. 6, no. 1, pp. 1-28, 1991.

Wong Lin Juan Linda is student from Malaysia. She studied computer science and software engineering for bachelor degree in Swinburne University, Sarawak Campus, graduating in the year 2013. She proceeded master of science degree in research in University Malaysia Sarawak (UNIMAS) in the year of 2015. She is currently in her last year of master degree, majoring in information system, natural language processing.

Chih How Bong completed master of computer science degree in University of Colorado Boulder in the year of 2009. He proceed to Ph.D in computer science from the same university , finished in 2011.

He is a senior lecturer at University Malaysia Sarawak, Malaysia. He is an academician with a demonstrated history of working in the software development and e-commerce. He has strong education professional with a Ph.D focused in natural language processing. His main interests are artificial intelligence, natural language processing, data science and machine learning.

Bong obtained 12 honors and awards to date and also a certification for "Train the Trainer" by Human Resources Development Fund Malaysia (HRDF).

Nung Kion Lee is a senior lecturer at the Faculty of Cognitive Sciences and Human Development, Universiti Malaysia Sarawak. He holds degrees in computer science in both of his undergraduate (UPM) and postgraduate studies (La Trobe University). He is the head of Research Cluster Intelligent System and Interactive Technology of the faculty. Lee has more than ten years of experience working on research in machine learning and optimization techniques. In machine learning, his research is concentrated on genomic-scale data for regulatory elements prediction. He has published considerably in that area. While in the optimization technique, he is an expert in genetic algorithm.