$TrWP$: Text Relatedness using Word and Phrase Relatedness

Md Rashadul Hasan Rakib **Aminul Islam** **Evangelos Milios**
Faculty of Computer Science
Dalhousie University, Canada
{rakib,islam,eem}@cs.dal.ca

Abstract

Text is composed of words and phrases. In bag-of-word model, phrases in texts are split into words. This may discard the inner semantics of phrases which in turn may give inconsistent relatedness score between two texts. $TrWP$, the unsupervised text relatedness approach combines both word and phrase relatedness. The word relatedness is computed using an existing unsupervised co-occurrence based method. The phrase relatedness is computed using an unsupervised phrase relatedness function f that adopts Sum-Ratio technique based on the statistics in the Google n-gram corpus of overlapping n-grams associated with the two input phrases. The second run of $TrWP$ ranked 30th out of 73 runs in SemEval-2015 task2a (English STS).

1 Introduction

Generally, a phrase is an ordered sequence of multiple words that all together refer to a particular meaning (Zamir and Etzioni, 1999). Phrase relatedness quantifies how two phrases relate to each other. It plays an important role in different Text Mining tasks; for instance, document similarity [1], classification and clustering are performed on the documents composed of phrases. Several document clustering methods use phrase similarity to determine the similarity between documents so as to improve the clustering result (Chim and Deng, 2008; Shrivastava et al., 2013). SpamED (Pera and Ng, 2009) uses the bi-gram and tri-gram phrase similarity between an incoming e-mail message and a previously marked spam to enhance the accuracy of spam detection.

Most works on text relatedness can be abstracted as a function of word relatedness (Ho et al., 2010). The classical Bag-of-Word (BoW) text relatedness methods split phrases into words; then compute text-pair relatedness by word-pair relatedness (Islam and Inkpen, 2008; Islam et al., 2012; Tsatsaronis et al., 2010). $TrWP$ considers text as Bag-of-Word-and-Phrase (BoWP). It considers a (word, bi-gram) or (bi-gram, bi-gram) pair as a phrase-pair [2] and computes text relatedness using both word and phrase relatedness.

There are phrase relatedness tasks that use compositional distributional semantic (CDS) model (Annesi et al., 2012; Hartung and Frank, 2011). Some use different tools and knowledge-based resources (Han et al., 2013; Tsatsaronis et al., 2010). These methods split phrases into words without considering the word order that might change the meaning of phrases leading to inconsistent phrase relatedness score (Turney and Pantel, 2010). For example, if we split the phrases "boat house" and "house boat" into words, we get the relatedness score one, nonetheless as a whole unit, these two phrases do not refer to exactly the same meaning (Turney and Pantel, 2010). To preserve the phrase meaning, $TrWP$ uses the phrase relatedness function f that considers a phrase as a single unit.

[1] We use 'relatedness' and 'similarity' interchangeably in our paper, albeit 'similarity' is a special case of 'relatedness'.

[2] We consider the bi-grams as phrases. A word is also considered as a phrase when relatedness is computed between word and bi-gram.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 90–95,
Denver, Colorado, June 4-5, 2015.

2 Terminology used in Phrase Relatedness

The terminologies used in measuring phrase relatedness are described below.

2.1 Bi-gram Context

Bi-gram context is a bi-gram, extracted by placing a phrase in the left most, middle and right position within the Google n(=3,4)-grams. Sample bi-gram contexts for the bi-gram phrase "large number" are shown in Table 1.

Phrase position	Google 4-grams
Left most	large number **of files**
Middle	**very** large number **generator**
Right most	**multiply a** large number

Table 1: Positions of the bi-gram phrase ("large number") in Google 4-grams and corresponding bi-gram contexts marked bold.

2.2 Overlapping Bi-gram Context

The overlapping bi-gram context is a bi-gram which is overlapped between two Google n(=3,4)-grams that contain two target phrases at the same position. Consider two Google 4-grams "large number of death" and "vast amount of death" where "large number" and "vast amount" are the target phrases and "of death" is an overlapping bi-gram context.

2.3 Sum-Ratio (SR)

Sum-Ratio refers to the product of sum and ratio between the minimum (min) and maximum (max) of two numbers. The Sum-Ratio of two numbers indicates the strength of association between them by maximizing the sum of two numbers with respect to their ratio. The objective of Sum-Ratio is to capture the strength of association between two overlapping Google n(=3,4)-grams. Given two numbers a and b, the Sum-Ratio of a and b is defined as follows.

$$Sum(a,b) = a + b$$
$$Ratio(a,b) = min(a,b)/max(a,b)$$
$$Sum\text{-}Ratio(a,b) = Sum \times Ratio$$

2.4 Relatedness Strength

Relatedness strength is the strength of association between two phrases P_1 and P_2, computed using the Sum-Ratio values between the counts of any two Google n(=3,4)-grams that contain P_1 and P_2, respectively and an overlapping bi-gram context.

3 Phrase Detection

Given a specific text, we elicit bi-grams of interest as candidate phrases if they are highly frequent in the Google bi-gram corpus, asserted in the Google Book-Ngram-Viewer (books.google.com/ngrams/info). We adopt a naive approach to detect the bi-gram phrases using the mean (u_{bg}) and standard deviation (sd_{bg}) of all Google bi-gram frequencies which are computed once. At first, the whole text is split by stop-words producing a list of c-grams [3]. Then for each c-gram, the following two steps are executed.

Step 1: If the c-gram is a bi-gram and its frequency is greater than $u_{bg} + sd_{bg}$, then we add it to the list of bi-gram phrases.

Step 2: If the length of c-gram is greater than two, we generate an array of bi-grams from the c-gram and find the most frequent bi-gram ($mfbg$) among them; If the frequency of $mfbg$ is greater than $u_{bg} + sd_{bg}$, then we add $mfbg$ to the list of bi-gram phrases and split the c-gram into two parts (e.g., left, right) by $mfbg$. After splitting, for each of the left and right parts, we examine the **Step 1** and **Step 2** recursively.

4 Computing Phrase Relatedness

The phrase relatedness function f, computes relatedness strength between two phrases P_1 and P_2 using the Google n-gram corpus (Brants and Franz, 2006) which is then normalized between 0 and 1 using NGD (Cilibrasi and Vitanyi, 2007) in conjunction with NGD´ (Gracia et al., 2006).

4.1 Lexical Pruning on the Bi-gram Contexts

At first the bi-gram contexts of phrases are extracted. However some phrases along with their bi-gram contexts do not convey meaningful insight due to the improper positioning of stop-words within bi-gram contexts. Therefore lexical pruning [4] is performed

[3] c-gram: A chunk of uni-grams with no stop-word.

[4] Perform pruning on the bi-gram contexts implies to the pruning of the Google n(=3,4)-grams from which those contexts are extracted.

based on the position of stop-words inside the bi-gram contexts. When the target phrase is placed at the left or right most positions respectively, then the Google n(=3,4)-gram is pruned if the right or left most word is a stop-word. When the phrase is in the middle surrounded by two context words, then the Google n(=3,4)-gram is pruned if both the surrounding context words are stop-words. After performing lexical pruning, we have two sets of non-pruned Google n(=3,4)-grams containing the bi-gram contexts of two phrases, respectively.

4.2 Finding Overlapping Bi-gram Contexts

We find the overlapping bi-gram contexts between two sets of non-pruned Google n(=3,4)-grams. The Google n(=3,4)-grams having overlapping bi-gram contexts are separated from the Google n(=3,4)-grams that have no overlapping contexts.

4.3 Statistical Pruning on the Overlapping Bi-gram Contexts

Each Google n(=3,4)-gram pair with overlapping bi-gram context possesses a strength of association. We presume that if most of the Google n(=3,4)-gram pairs have higher strengths of association, the relatedness score between two phrases tends to be higher and vice versa. However some strengths of association do not lie within the group of maximum number of strengths of association called outliers and because of the outliers the relatedness score between two phrases becomes inconsistent. Hence we apply statistical pruning on the strengths of association to prune the outliers. To find the group of maximum number of strengths of association and prune the outliers, we adopt the Normal Distribution (Bohm and Zech, 2010) for statistical pruning. It has been shown that in Normal Distribution most of the samples exist within the mean $\pm$ standard deviation.

We divide each Google n(=3,4)-gram count (frequency) within a pair by the count of its corresponding n(=1,2)-gram phrase, resulting a normalized count. For each Google n(=3,4)-gram pair, the minimum and maximum among the two normalized counts are determined. After that we calculate the ratio (e.g., minimum/maximum) between them. Following that, for each Google n(=3,4)-gram pair, we multiply the ratio with the sum of two Google n(=3,4)-gram counts, producing a resultant product (e.g., strength of association). Later on we compute the mean (u_{sr}) and standard deviation (sd_{sr}) from the strengths of association of the Google n(=3,4)-gram pairs. If the strength of association is within the $u_{sr} \pm sd_{sr}$, it is kept otherwise pruned.

4.4 Computing Relatedness Strength

Relatedness strength between P_1 and P_2 is computed by multiplying the relatedness strengths from overlapping and all bi-gram contexts.

4.4.1 Relatedness Strength using Overlapping Bi-gram Contexts

For each non-pruned Google n(=3,4)-gram pair having overlapping bi-gram context, the strength of association is calculated following the Sum-Ratio technique. We sum the two Google n(=3,4)-gram counts and find the minimum and maximum among them. After that we calculate the ratio (e.g., minimum/maximum) between them. Then the Sum-Ratio value is calculated by multiplying the sum with ratio which signifies the strength of association for a Google n(=3,4)-gram pair. By summing up the strength of association of each Google n(=3,4)-gram pair, we get the relatedness strength between the phrases P_1 and P_2 denoted by $RSOB(P_1, P_2)$ as shown in Eq. 1. GP_1 and GP_2 are the Google n(=3,4)-grams that contain P_1 and P_2, respectively and an overlapping bi-gram context. $C(GP_1)$ and $C(GP_2)$ are the counts of GP_1 and GP_2, respectively. k is the number of non-pruned Google n(=3,4)-gram pairs.

$$RSOB(P_1, P_2) = \sum^{n} \frac{min(C(GP_1), C(GP_2))}{max(C(GP_1), C(GP_2))} \times sum(C(GP_1), C(GP_2)) \quad (1)$$

4.4.2 Relatedness Strength using all Bi-gram Contexts

All bi-gram contexts of a phrase P_1 include both non-pruned overlapping and non-overlapping bi-gram contexts, extracted from the Google n(=3,4)-grams where P_1 appears. Two vectors V_1 and V_2 in Vector Space Model are constructed for P_1 and P_2, respectively using their corresponding all bi-gram Contexts. The elements of V_1 and V_2 are binary and reflect the presence or absence of a bi-gram

context belonging to the phrases P_1 and P_2, correspondingly. The relatedness strength between P_1 and P_2 using all bi-gram contexts is designated as $cosSim(V_1, V_2)$, and computed by the cosine similarity between V_1 and V_2, defined in Eq. 2.

$$cosSim(V_1, V_2) = \frac{V_1.V_2}{||V_1||\ ||V_2||} \tag{2}$$

4.4.3 Multiplying Relatedness Strengths from Overlapping and all Bi-gram Contexts

We multiply the relatedness strengths $RSOB(P_1, P_2)$ and $cosSim(V_1, V_2)$ obtained from overlapping and all bi-gram contexts, respectively to compute the overall relatedness strength $f(P_1, P_2)$ between the phrases P_1 and P_2, defined in Eq. 3. The purpose of multiplying these two strengths is to quantify $RSOB(P_1, P_2)$ with respect to $cosSim(V_1, V_2)$.

$$f(P_1, P_2) = RSOB(P_1, P_2) \times cosSim(V_1, V_2) \tag{3}$$

4.5 Normalizing Overall Relatedness Strength

The relatedness between phrases P_1 and P_2 is computed by normalizing the overall relatedness strength between 0 and 1 using NGD in conjunction with NGD´ as defined in Eq. 4. $C(P)$ is the count of phrase P where P is a Google n(=1,2)-gram. N = total number of web documents used in the Google n-gram corpus.

$$NGDf(P_1, P_2) = e^{-2\times\frac{max(log\ \ C(P_1), log\ \ C(P_2)) - log\ \ f(P_1,P_2)}{log\ \ N - min(log\ \ C(P_1), log\ \ C(P_2))}} \tag{4}$$

5 Computing Text Relatedness

At first punctuations are removed from texts. The phrases are extracted using phrase detection algorithm. Other than phrases the rest of the text is split into non stop-words. The relatedness between two texts is calculated by the word-pair and phrase-pair relatedness following the notion of text relatedness in (Islam et al., 2012). Word-pair relatedness is computed by the word relatedness method in (Islam et al., 2012).

Step 1: We assume that the two texts $A = \{a_1, a_2, ..., a_p\}$ and $B = \{b_1, b_2, ..., b_q\}$ have p and q tokens, respectively and $p \leq q$. Otherwise we switch A and B. A token is a word or bi-gram phrase.

Step 2: We count the number of common tokens (δ) in both A and B where $\delta \leq p$. Common tokens are determined by applying PorterStemmer (Porter, 1980) on each token pair. Common tokens are removed from A and B. So, $A = \{a_1, a_2, ..., a_{p-\delta}\}$ and $B = \{b_1, b_2, ..., b_{q-\delta}\}$. If all tokens match e.g., $p - \delta = 0$, go to step **Step 5**.

Step 3: We construct a $(p - \delta) \times (q - \delta)$ 'semantic relatedness matrix' (Say, $M = (\alpha_{ij})_{(p-\delta)\times(q-\delta)}$) using the following process. We set $\alpha_{ij} \leftarrow relatedness(a_i, b_j) \times w^2$ where $i = 1...p - \delta$, $j = 1...q - \delta$, w = weighting factor to boost the relatedness score. The value of w is the average number of words within a word or phrase-pair. The reason for boosting is that same relatedness score of a phrase-pair is more weighted than that of a word-pair. If (a_i, b_j) is a word-pair, $relatedness(a_i, b_j)$ = word-pair relatedness (Islam et al., 2012); otherwise $relatedness(a_i, b_j)$ = phrase-pair relatedness from Eq. 4.

$$M = \begin{pmatrix} \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,q-\delta} \\ \alpha_{2,1} & \cdots & \alpha_{2,j} & \cdots & \alpha_{2,q-\delta} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{i,1} & \cdots & \alpha_{i,j} & \cdots & \alpha_{i,q-\delta} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{p-\delta,1} & \cdots & \alpha_{p-\delta,j} & \cdots & \alpha_{p-\delta,q-\delta} \end{pmatrix}$$

Step 4: For each row we compute the mean (u) and standard deviation (sd) of the relatedness scores and select the scores which are larger than $u+sd$. The idea is to find more related tokens among $(q - \delta)$, for each $(p - \delta)$ tokens. The average of the selected scores is computed for a row and for $(p-\delta)$ rows we get $(p - \delta)$ averages. We sum the $(p - \delta)$ average values denoted by $SAvg$.

Step 5: To compute relatedness between the texts A and B, we use the normalization in (Islam et al., 2012) with minor modification, given in Eq. 5.

$$rel.(A, B) = \frac{(2|\delta| + SAvg) \times (2|A| + 2|B|)}{2 \cdot 2|A| \cdot 2|B|} \tag{5}$$

Number of words in A, B and δ are denoted by $|A|$, $|B|$, $|\delta|$, respectively. Since we multiply w with relatedness score while constructing the matrix M; $|A|$, $|B|$ and $|\delta|$ are multiplied by 2.

6 Experiments

We submit three runs of $TrWP$ on 5 datasets of SemEval-2015 task2a (English STS) (Agirre et al., 2015).

6.1 Run1

In the first run we consider words, phrases and numbers as tokens. After removing punctuations and stop-words, if any sentence within a pair has no tokens, then the relatedness of that sentence pair is 0.

6.2 Run2

The tokens are same as in the first run. After removing punctuations and stop-words, if any sentence within a pair has no tokens, then we keep the stop-words.

6.3 Run3

We consider words and phrases as tokens. The following steps are same as in the first run.

7 Result

The result from three different runs of $TrWP$ are shown in Table 2.

SemEval-2015 task2a Dataset (English STS)	Run1 (r)	Run2 (r)	Run3 (r)
answers-forums	0.6857	0.6857	0.6857
answers-students	0.6618	0.6618	0.6612
belief	0.6769	0.7245	0.6772
headlines	0.7709	0.7709	0.7710
images	0.7865	0.7865	0.7865
Weighted mean	0.7251	0.7311	0.7250
Ranking out of 73 runs	31	30	32

Table 2: Pearson's r on five datasets, obtained from three different runs of $TrWP$.

8 Conclusion

$TrWP$ is an unsupervised text relatedness method that combines both word and phrase relatedness. Both the word and phrase relatedness are computed in unsupervised manner. The word relatedness is computed using the co-occurrences of two words in the Google 3-gram corpus. To compute phrase relatedness, $TrWP$ uses an unsupervised function f based on the Sum-Ratio technique along with the statistical pruning. Unlike other phrase relatedness methods based on word relatedness, f considers the whole phrase as a single unit without losing inner semantic meaning within a phrase.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, CO, June.

Paolo Annesi, Valerio Storch, and Roberto Basili. 2012. Space projections as distributional models for semantic composition. In *Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I*, CICLing'12, pages 323–335, Berlin, Heidelberg.

Gerhard Bohm and Gnter Zech. 2010. *Introduction to statistics and data analysis for physicists*.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram corpus version 1.1. *Linguistic Data Consortium*.

Hung Chim and Xiaotie Deng. 2008. Efficient phrase-based document similarity for clustering. *IEEE Trans. on Knowl. and Data Eng.*, 20(9):1217–1229, September.

Rudi L. Cilibrasi and Paul M. B. Vitanyi. 2007. The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):370–383, March.

Jorge Gracia, Raquel Trillo, Mauricio Espinoza, and Eduardo Mena. 2006. Querying the web: A multiontology disambiguation method. In *Proceedings of the 6th International Conference on Web Engineering*, ICWE '06, pages 241–248, New York, NY, USA.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Johnathan Weese. 2013. UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, June.

Matthias Hartung and Anette Frank. 2011. Assessing interpretable, attribute-related meaning representations for adjective-noun phrases in a similarity prediction task. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '11, pages 52–61, Stroudsburg, PA, USA.

Chukfong Ho, Masrah Azrifah Azmi Murad, Rabiah Abdul Kadir, and Shyamala C. Doraisamy. 2010. Word sense disambiguation-based sentence similarity. In *Proceedings of the 23rd International Conference on*

Computational Linguistics: Posters, COLING '10, pages 418–426, Stroudsburg, PA, USA.

Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data*, 2(2):10:1–10:25, July.

Aminul Islam, Evangelos Milios, and Vlado Kešelj. 2012. Text similarity using google tri-grams. In *Proceedings of the 25th Canadian conference on Advances in Artificial Intelligence*, Canadian AI'12, pages 312–317, Berlin, Heidelberg.

Maria Soledad Pera and Yiu-Kai Ng. 2009. Spamed: A spam e-mail detection approach based on phrase similarity. *J. Am. Soc. Inf. Sci. Technol.*, 60(2):393–409, February.

Martin Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137, July.

Shailendra Kumar Shrivastava, J. L. Rana, and R. C. Jain. 2013. Article: Text document clustering based on phrase similarity using affinity propagation. *International Journal of Computer Applications*, 61(18):38–44, January. Published by Foundation of Computer Science, New York, USA.

George Tsatsaronis, Iraklis Varlamis, and Michalis Vazirgiannis. 2010. Text relatedness based on a word thesaurus. *J. Artif. Int. Res.*, 37(1):1–40, January.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January.

Oren Zamir and Oren Etzioni. 1999. Grouper: A dynamic clustering interface to web search results. In *Proceedings of the Eighth International Conference on World Wide Web*, WWW '99, pages 1361–1374, New York, NY, USA.