Integrated Sentiment and Emotion into Estimating the Similarity Among Entries on Social Network

Thi Hoi Nguyen[1], Dinh Que Tran[2], Gia Manh Dam[1], and Manh Hung Nguyen[2,3](✉)

[1] Vietnam Commercial University, Hanoi, Vietnam
hoint2002@gmail.com, damgiamanh@gmail.com

[2] Posts and Telecommunications Institute of Technology (PTIT), Hanoi, Vietnam
tdque@yahoo.com, nmh.nguyenmanhhung@gmail.com

[3] UMI UMMISCO 209 (IRD/UPMC), Hanoi, Vietnam

Abstract. Similar measures play an important role in information processing and have been widely investigated in computer science. With the exploration of social media such as Youtube, Wikipedia, Facebook etc., a huge number of entries have been posted on these portals. They are often described by means of short text or sets of words. Discovering similar entries based on such texts has become challenges in constructing information searching or filtering engines and attracted several research interests. In this paper, we firstly introduce a model of entries posted on media or entertainment portals, which is based on their features composed of title, category, tags, and content. Then, we present a novel similar measure among entries that incorporates their features. The experimental results show the superiority of our incorporation similarity measure compared with the other ones.

Keywords: Similar measure · Social network · Text · Entry
Social media

1 Introduction

Recently, the exploration of social networks attracts not only the user to participle, but also many of researchers to mining and benefit the huge amount of data posted in these social networks. The entries posted are often included short text or sets of words to describe viewpoints, comments and so on. Discovering similar entries based on such texts has become challenges in constructing information searching or filtering engines and attracted several research interests.

The problem of how to detect the similarity between two objects, in general, has been investigated for decades (Lin [1], Sayal and Kumar [2], Reddy and Krishnaiah [3], Nguyen and Nguyen [4]). However, these models are too general to be applied into estimating the similarity among entries posted in social networks.

Y. Chen and T. Q. Duong (Eds.): INISCOM 2017, LNICST 221, pp. 242–253, 2018.
https://doi.org/10.1007/978-3-319-74176-5_21

A closer approach to the problem is to use models to detect the similarity among texts, including the models based on semantic such as Buscaldi et al. [5], or Han et al. [6], Lee et al. [7], Marsi et al. [8], Oliva et al. [9], Agirre et al. [10], Nguyen and Tran [11,12], Novelli and Oliveira [13]; or based on statistic method such as Bollegala et al. [14], Buscaldi et al. [15], Croce et al. [16], Finkel et al. [17], Lintean and Rus [18], Proisl et al. [19], Saric et al. [20], Severyn et al. [21], Sultan et al. [22], Xu and Lu [23]. However, most of these models consider only the text body to estimate their similarity. They lack of investigating additional information such as the tags, category, title, keywords, sentiment, and emotion which may contribute greatly to estimate the similarity of entities.

On the line with our previous work (Nguyen et al. [24]) which considered three features of entries (*content*, *category*, *tags*), this paper integrates two more features to distinguish entries: *sentiment* and *emotion* implicitly presented in the entries. Experiments will be performed to validate and evaluate the performance of our model compared with other ones.

In order to see the role of *sentiment* and *emotion* in distinguishing entries, let's consider three following entries (which are extracted from Twitter - www.twitter.com):

- A: *Gone for a run beautiful morning man do I love iOS 5 @apple #iPhone*
- B: *I hate my apple computer. Thats 3500 dollars down the drain.*
- C: *Thank you @apple for Find My Mac - just located and wiped my stolen Air.*

If there ware only three features of *content*, *category*, and *tags* considered, it could be difficult to conclude whether the entry B or the entry C is more similar to the entry A than the other because all three entries say about the same category (*technology*), the same tags (*apple, iPhone, Mac, iOS*). However, if we take into account the features of *sentiment* and *emotion*, it is more easily than before to conclude that the entry C is more similar to the entry A than the entry B is: the entry A and C have the same *sentiment* (positive), meanwhile the entry B has a negative value of *sentiment*. And, at the level of *emotion*, the entry A may have *love, joy*, that more or less close to the *gratitude* emotion from the entry C, and that is contrary the *disgust* emotion from the entry B.

Someone could argues why do we need to consider both *sentiment* and *emotion* of entries while the sentiment could be inferred from emotion such as: positive emotion brings positive sentiment, and vice versa. However, in reality, the answer is yes, we do. The *sentiment* is the opinion of the user about the topic in the entry. Meanwhile the *emotion* is the emotional status of the user in the entry. Consequently, in many cases, *sentiment* is independent from the *emotion* in an entry. Let's consider these following entries (which are also extracted from Twitter):

- D: *@Mayati I think @Apple did not do such a thorough job with the step x steps for upgrade and move to iCloud.*
- E: *just like a coin has 2 sides, everyone has 2 faces...?*

– F: *@azee1v1 @apple @umber AppStore is well done so is iTunes on the mobile devices. I was talking about desktop app.*

It is easy to see that the entry D has negative *sentiment*, but no *emotion*. The entry E has neutral value of *sentiment*, but has confused in *emotion*. The entry F has positive *sentiment* but has no *emotion*. These examples indicate that the *sentiment* and *emotion* of an entry sometimes could be independent from each other. That's why we need to consider both these features in distinguishing the entries. The empirical results in our experiment also indicate the importance of both these two features.

The paper is organized as follows. Section 2 presents the model of entries and their similarity measure based on similarities of features. Section 3 describes experiments to evaluate the proposed model. Section 4 is the conclusion and perspectives.

2 A Similarity Measure Model for Social Network Entries

The general model takes the two entries as input data and the output is the estimated similarity between the two entered entries. Inside the model, there are four main steps:

– Step 1: Modeling entries.
– Step 2: Extracting the value for *implicit attribute* of entries.
– Step 3: Estimating the similarity on each entry attribute.
– Step 4: Aggregating the similarity between entries from their similarities on attributes.

These steps will be described in detail in the next sections.

2.1 Modeling Entries

Without loss of generality, we assume that:

– An entry on a social network could be: a text, an image, an audio stream, a video stream, or a combination of these medias. In this model, we consider only the textual part in an entry. Therefore, an entry could be considered as a text.
– An entry could be originally posted by an user, or shared (referred) from another user or another online source. This model consider an entry is the whole text, including the directly posted text, and the text referred from other source.
– An entry could have several attributes, including *explicit* attributes such as the *content*, and the *implicit* attributes such as: category, sentiment, emotion. As the *implicit* attributes could not directly extracted from an entry, the model needs a step to extract these attributes before estimating the similarity on them. This model consider five attributes of an entry:

- *Content* of entry i, noted as f^i_{con}: is the whole text part in the entry itself. This is an *explicit* attribute.
- *Tags* of entry i, noted as f^i_{tag}: An entry could be tagged to a set of tags. Each tag is an independent word or expression. In some case, tags could be directly tagged by the user (explicit). In some other case, it is not explicitly tagged by the user (implicit).
- *Category* of entry i, noted as f^i_{cat}: An entry could be assigned to a category. Each category is represented by an independent word or expression.
- *Sentiment* of entry i, noted as f^i_{sen}: An entry could have a sentiment of the user. A sentiment be in agree (positive), disagree (negative), or neutral opinion.
- *Emotion* of entry i, noted as f^i_{emo}: An entry could also have some emotion of the user. Each emotion is represented by an independent word or expression.

As an entry is considered as a set of attributes and only their textual values are considered. And then the problem of estimating the similarity among entries becomes the computation of the similarity among texts or among sets of expressions.

2.2 Auto Extract Value for Implicit Attribute of Entry

Let's consider an example of a status on Twitter: "*Thank you @apple for Find My Mac - just located and wiped my stolen Air*". When we see this status, only the *content* is explicitly presented, that is the whole text of the status. However, we could quickly identify some other attributes of this status, such as *category* could be (*technology*), *tags* could be (*apple, Mac*), *sentiment* could be (*neutral* - neither agree nor disagree), and *emotion* could be (*gratitude, joy*). The attributes whose value is not explicitly presented in the entry but it could be extracted from the inside of the entry are called *implicit attributes*. Our objective in this step is to extract the value of *implicit attributes* of an entry.

In order to do this, we could apply any existed supervised machine learning method. In this model, we apply a method to extract value of each of four implicit attributes as follow:

- Step 1: Construct a set of labeled samples (texts), called *training set*. In which, each text is assigned to a set of labels. The union of all labels of all texts called the set of labels L.
- Step 2: For each label $l_i \in L$, create two sets of text sample:
  - T_{l_i} is the set of all texts which are labeled with the label l_i.
  - $T_{\neg l_i}$ is the set of all texts which are not labeled with the label l_i.
- Step 3: For each text $t_k \in T_{l_i}$ $(T_{\neg l_i})$, calculate the label oriented features as follow:
  - Split t_i into a set of n-gram or term (stop words could be removed).
  - Take the union of all terms in all texts in the set T_{l_i} and $T_{\neg l_i}$

- Calculate the *label oriented term score* of each term in the corresponding set for each label l_i:

$$s_{LOT}(x, l_i) = \frac{N^x_{l_i}}{N_{l_i}} * log\left(\frac{N_{\neg l_i}}{N^x_{\neg l_i}}\right) - \frac{N^x_{\neg l_i}}{N_{\neg l_i}} * log\left(\frac{N_{l_i}}{N^x_{l_i}}\right) \tag{1}$$

where, $N_l, N_{\neg l}$ are the number of text in the set T_l, $T_{\neg l}$, respectively. $N^x_l, N^x_{\neg l}$ are the number of text in the set T_l, $T_{\neg l}$, respectively, which contains the term x.

- Step 4: For a new text t, the choice of label to assign to the text is follow:
  - Split t into a set of n-grams or terms $X = (x_1, x_2, ...x_n)$.
  - Calculate the *term frequency* for each term x_i in the text t: $tf(x_i, t)$.
  - For each label $l_i \in L$, calculate the *label oriented document score*:

$$s_{LOD}(t, l_i) = \frac{1}{n_t} * \sum_{x \in t} s_{LOT}(x, l_i) * tf(x, t) \tag{2}$$

  - If $s_{LOD}(t, l_i) > 0$:
    - In the multi-label problem where a text could be assigned to several labels, the text t will be labeled with the label l_i.
    - In the single label problem where a text could be assigned to only one label, it is needed to calculate all the final label oriented (disoriented) scores of the text t for the all labels $l_i \in L$. And the label whose *label oriented document score* is the highest score will be assigned to the text t.

2.3 Similarity on Each Attribute

As only textual value of feature is considered, we distinguish two kinds of textual value of feature:

- First, the feature value is already in form of a set of expressions, such as the value of feature *tag*, *category*, *sentiment*, and *emotion*. Their similarity is resulted to considered among sets of expressions.
- Second, the feature value is in form of a general text, such as the value of feature *content*. Their similarity is considered among texts.

Attribute Whose Value is a Set of Expressions. Since the attribute value is in the form of a set of textual expressions, their similarity is defined as follows:

Suppose that $A_1 = (a^1_1, a^2_1, ...a^m_1)$, $A_2 = (a^1_2, a^2_2, ...a^n_2)$ are two sets of expressions or strings, where m, n are the sizes of the set A_1 and A_2, respectively. Let v be the size of the intersection of A_1 and A_2. The similarity between A_1 and A_2 is defined by the formula:

$$s_{exp}(A_1, A_2) = \frac{2* \mid A_1 \cap A_2 \mid}{\mid A_1 \mid + \mid A_2 \mid} = \frac{2 * v}{m + n} \tag{3}$$

It is clear that all possible values of $s_{exp}(A_1, A_2)$ are lied in the interval $[0, 1]$. This formula could be applied to the attributes whose value is a set of expressions.

Suppose that $i = (f_1^i, f_2^i, ..f_n^i)$, $j = (f_1^j, f_2^j, ..f_n^j)$ are two entries represented by their attributes. Let consider the attribute k whose value is a set of expressions. The similarity between entries i and j on the attribute k is defined by the formula:

$$s_k(i, j) = s_{exp}(f_k^i, f_k^j) \tag{4}$$

where f_k^i, f_k^j are the expression values of the attribute k of the two entries i and j. For examples, the similarity on attribute *category*, *tags*, *sentiment*, and *emotion* of two entries are given as follows:

$$s_{cat}(i, j) = s_{exp}(f_{cat}^i, f_{cat}^j) \tag{5}$$

$$s_{tag}(i, j) = s_{exp}(f_{tag}^i, f_{tag}^j) \tag{6}$$

$$s_{sen}(i, j) = s_{exp}(f_{sen}^i, f_{sen}^j) \tag{7}$$

$$s_{emo}(i, j) = s_{exp}(f_{emo}^i, f_{emo}^j) \tag{8}$$

Attribute Whose Value is a Text. In this case, the problem becomes the estimation the similarity between two texts. We could apply the technique TF-IDF (Term Frequency - Inverse Document Frequency) [25] to characterize the texts, which are used in many statistic-based models such as Buscaldi et al. [15], Finkel et al. [17]. In our work, TF-IDF is also used to estimate the similarity between two features of text value as follows:

- Extract the attribute value (a text) into a set of n-gram $t^1 = (g_1^1, g_2^1, ...g_n^1)$ and $t^2 = (g_1^2, g_2^2, ...g_m^2)$
- Calculate the TF-IDF of each n-gram in the text. Then represent the attribute value by a vector whose each element is a pair $<$ *n-gram, td-idf* $>$: $v^1 = (< g_1^1, v_1^1 >, < g_2^1, v_2^1 >, ... < g_n^1, v_n^1 >)$ and $v^2 = (< g_1^2, v_1^2 >, < g_2^2, v_1^2 >, ... < g_m^2, v_m^2 >)$
- Calculate the distance between the two vectors:

$$D(v^1, v^2) = \frac{1}{N} \sum_{1}^{N} d_k \tag{9}$$

  where N is the number of different n-grams considered in both $t^1 \cup t^2$, d_k is the distance on each element $< g_i^1, v_i^1 >$ of v^1 (or element $< g_j^2, v_j^2 >$ of v^2, respectively):
  - If there is an element $< g_l^2, v_l^2 >$ of v^2 (or element $< g_l^1, v_l^1 >$ of v^l, respectively) such that $g_l^2 = g_i^1$, then:

$$d_k = \frac{| v_i^1 - v_l^2 |}{max(v_i^1, v_l^2)} \tag{10}$$

  - Otherwise, $d_k = 1$.

- It is clear that the value of $D(v^1, v^2)$ is in the interval $[0, 1]$. Similarity between the two features is then:

$$s_{txt}(t^1, t^2) = 1 - D(v^1, v^2) \tag{11}$$

For example, similarity between the attribute *content* of two entries i and j is as follows:

$$s_{con}(i, j) = s_{txt}(f^i_{con}, f^j_{con}) \tag{12}$$

2.4 Similarity Between Two Entries

Once the similarities between two entries on each attribute are estimated, the similarity between two entries is then computed by a weighted average aggregation of the similarity on all considered attributes as follows.

Suppose that $i = (f^i_1, f^i_2, ..f^i_n)$, $j = (f^j_1, f^j_2, ..f^j_n)$ are two entries represented by their attributes. The similarity between entries i and j on all considered attributes is defined by the formula:

$$s_{entry}(i, j) = \sum_{k=1}^{n} w_k * s_k(i, j) \tag{13}$$

where $s_k(i, j)$ is the similarity on attribute k of entries i and j; w_k is the weight of the feature k such that:

$$\sum_{k=1}^{n} w_k = 1 \tag{14}$$

The more this similarity is closed to 1, the more the two entries are similar. And vice versa, the more this similarity is closed to 0, the less the two entries are similar.

3 Experimental Evaluation

This section first describes the construction of sample set and then presents the experiments and evaluation results.

3.1 Construction of Sample Set

In order to have entries which have sentiment and emotions, we collected more than 1000 statuses posted on Twitter (twitter.com). Samples are then constructed from these Twitter statuses. Each sample contains:

- The *id* of the sample.
- The *value* of the sample. It could be 1 or 2.

- Each sample contains three entries collected from Twitter. These entries are called as entry A, B, and C. And the *value* of the sample is determined as follow:
  - If the entry A is more similar to the entry B than C, then the *value* of this sample is 1.
  - In the contrary, if the entry A is more similar to the entry C than B, then the *value* of this sample is 2.

In this experiment, we constructed and use 500 samples. For instance, a sample is presented in Table 1. In this sample, the entry A is similar to the entry C than the entry B, so the *value* of the sample is 2.

Table 1. An example of a sample constructed from Twitter entries

ID	354
Value	2
A	Gone for a run beautiful morning man do I love iOS 5 @apple #iPhone
B	@AsimRang @apple @umber the desktop app is wack though
C	Thank you @apple for Find My Mac - just located and wiped my stolen Air

3.2 Method of Experiment and Evaluation

In order to compare the results of our model to other related works, we choose and implement these following models:

- *Model 1*: It is the model of Buscaldi et al. [15] which takes into account only the *content* attribute of entries.
- *Model 2*: It is the model of Nguyen et al. [24] which takes into account three attributes of entries: *content*, *category*, and *tags*.
- *Model 3*: It is our model which takes into account five attributes of entries: *content*, *category*, *tags*, *sentiment*, and *emotion*. In this model, we use 3 values of *sentiment*, and 16 values of *emotion* as indicated in Table 2. These emotions are detected based on the cognitive definition of emotion proposed by Ortony et al. [26], Frijda [27], and Reisenzei [28].

The experiment is performed as follows on each model:

- For each sample, we use model proposed in this paper to estimate the similarity between the entry B and A, and that between entry C and A.
- If A is more similar to B than C is, then the *result* of this sample is 1. In the contrary, If A is more similar to C than B is, then the *result* of this sample is 2.

Table 2. Values of *sentiment* and *emotion* used in the Model 3

Attribute	Values
Sentiment	Positive
	Negative
	Neutral
Emotion	Joy
	Sad
	Happyfor
	Sorry
	Hope
	Fear
	Regret
	Disappointed
	Love
	Disgust
	Confused
	Pride
	Anger
	Gratitude
	Admiration
	No emotion

– We then compare the *result* and the *value* of each sample. If they are identical, we increase the variable *number of correct sample* by 1.

In order to evaluate the results, we make use of the correct ratio (CR) of the model over the given sample set which is calculated as follows:

$$CR = \frac{\text{number of correct sample}}{\text{total of sample}} * 100\% \tag{15}$$

The more the CR value is closed to 100%, the more the model is correct. We expect that the obtained value of CR is high as much as possible.

3.3 Results

The results are presented in the Table 3. They indicate that our model, which reaches the correct ratio of 86.20%, is significantly better than the model of Buscaldi et al. [15] (with CR = 69.00%) and Nguyen et al. [24] (with CR = 79.40%), regarding the given sample set.

The results also determined the best combination of attribute weights for each model. Meanwhile the model of Buscaldi et al. [15] concentrate 100% on the *content* so there is no option to choose the best. The model of Nguyen et al. [24] considered only three attributes *content*, *category*, and *tag*, so the best combination of weights is 0.65 : 0.20 : 0.15, respectively. The best combination of

Table 3. Results of considered models

Model	CR (%)	Best weight combination				
		w_1	w_2	w_3	w_4	w_5
Buscaldi et al. [15]	69.00	1				
Nguyen et al. [24]	79.40	0.65	0.20	0.15		
Our model	86.20	0.30	0.30	0.05	0.05	0.30

five weights corresponding five attributes *content*, *category*, *tag*, *sentiment*, and *emotion* in our model is 0.30 : 0.30 : 0.05 : 0.05 : 0.30, respectively. This results also indicate the role of *sentiment* and *emotion* in differentiating the entries in Twitter.

4 Conclusions

This paper presented the integration of two new attributes, *sentiment* and *emotion*, into the considered attribute set to estimate the similarity among entries in social networks. The model is then validated with empirical data collected from Twitter. The experimental results indicate that the proposed model could reach a higher value in accuracy than some recent related models.

Currently, we are considering how to take the semantic of text into account to compare expressions. These research results will be presented in our future work.

References

1. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning, pp. 296–304. Morgan Kaufmann, San Francisco (1998)
2. Sayal, R., Kumar, V.V.: A novel similarity measure for clustering categorical data sets. Int. J. Comput. Appl. **17**(1), 25–30 (2011). Published by Foundation of Computer Science
3. Reddy, G.S., Krishnaiah, R.V.: A novel similarity measure for clustering categorical data sets. IOSR J. Comput. Eng. (IOSRJCE) **4**(6), 37–42 (2012)
4. Nguyen, M.H., Nguyen, T.H.: A general model for similarity measurement between objects. Int. J. Adv. Comput. Sci. Appl. (IJACSA) **6**(2), 235–239 (2015)
5. Buscaldi, D., Roux, J.L., Flores, J.J.G., Popescu, A.: Lipn-core: Semantic text similarity using n-grams, wordnet, syntactic analysis, esa and information retrieval based features (2013)
6. Han, L., Kashyap, A.L., Finin, T., Mayfield, J., Weese, J.: Semantic textual similarity systems. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 44–52. Association for Computational Linguistics, Atlanta, June 2013

7. Lee, M.C., Chang, J.W., Hsieh, T.C.: A grammar-based semantic similarity algorithm for natural language sentences. Sci. World J. **2014**, 17 (2014)
8. Marsi, E., Moen, H., Bungum, L., Sizov, G., Gambäck, B., Lynum, A.: Combining strong features for semantic similarity. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 66–73. Association for Computational Linguistics, Atlanta, June 2013
9. Oliva, J., Serrano, J.I., del Castillo, M.D., Iglesias, Á.: Symss: a syntax-based measure for short-text semantic similarity. Data Knowl. Eng. **70**(4), 390–405 (2011)
10. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W.: Semantic textual similarity. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 32–43. Association for Computational Linguistics, Atlanta, June 2013
11. Nguyen, M.H., Tran, D.Q.: A semantic similarity measure between sentences. South-East Asian J. Sci. **3**(1), 63–75 (2014)
12. Tran, D.Q., Nguyen, M.H.: A mathematical model for semantic similarity measures. South-East Asian J. Sci. **1**(1), 32–45 (2012)
13. Novelli, A.D.P., Oliveira, J.M.P.D.: Article: a method for measuring semantic similarity of documents. Int. J. Comput. Appl. **60**(7), 17–22 (2012)
14. Bollegala, D., Matsuo, Y., Ishizuka, M.: A web search engine-based approach to measure semantic similarity between words. IEEE Trans. Knowl. Data Eng. **23**(7), 977–990 (2011)
15. Buscaldi, D., Rosso, P., Gomez-Soriano, J.M., Sanchis, E.: Answering questions with an n-gram based passage retrieval engine. J. Intell. Inf. Syst. **34**(2), 113–134 (2010)
16. Croce, D., Storch, V., Basili, R.: Combining text similarity and semantic filters through sv regression. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 59–65. Association for Computational Linguistics, Atlanta, June 2013
17. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL 2005, pp. 363–370. Association for Computational Linguistics, Stroudsburg (2005)
18. Lintean, M.C., Rus, V.: Measuring semantic similarity in short texts through greedy pairing and word semantics. In: Youngblood, G.M., McCarthy, P.M. (eds.) Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference, Marco Island, Florida, 23–25 May 2012. AAAI Press (2012)
19. Proisl, T., Evert, T., Greiner, P., Kabashi, B.: Robust semantic similarity at multiple levels using maximum weight matching. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 532–540. Association for Computational Linguistics and Dublin City University, Dublin, August 2014
20. Šarić, F., Glavaš, G., Karan, M., Šnajder, J., Bašić, B.D.: Takelab: systems for measuring semantic text similarity. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, SemEval 2012, pp. 441–448. Association for Computational Linguistics, Stroudsburg (2012)

21. Severyn, A., Nicosia, M., Moschitti, A.: Tree kernel learning for textual similarity. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 53–58. Association for Computational Linguistics, Atlanta, June 2013
22. Sultan, M.A., Bethard, S., Sumner, T.: Sentence similarity from word alignment. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 241–246. Association for Computational Linguistics and Dublin City University, Dublin, August 2014
23. Xu, J., Lu, Q.: Computing semantic textual similarity using overlapped senses. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 90–95. Association for Computational Linguistics, Atlanta, June 2013
24. Nguyen, T.H., Tran, D.Q., Dam, G.M., Nguyen, M.H.: Multi-feature based similarity among entries on media portals. In: Akagi, M., Nguyen, T.-T., Vu, D.-T., Phung, T.-N., Huynh, V.-N. (eds.) ICTA 2016. AISC, vol. 538, pp. 373–382. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-49073-1_41
25. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill Inc., New York (1986)
26. Ortony, A., Clore, G.L., Collins, A.: The Congnitive Structure of Emotions. The Cambridge University Press, Cambridge (1988)
27. Frijda, N.H.: The Emotions: Studies in Emotion & Social Interaction. Edition de la Maison des Sciences de l'Homme, ser edn. Cambridge University Press, Paris (1986)
28. Reisenzein, R.: Emotions as metarepresentational states of mind: Naturalizing the belief-desire theory of emotion. Cognit. Syst. Res. **10**(1), 6–20 (2009)