Reflective Writing Analysis Approach Based on Semantic Concepts: An Evaluation of WordNet Affect Efficiency

Huda Alrashidi(✉) and Mike Joy

University of Warwick, Coventry, UK
{H.alrashidi,m.s.joy}@warwick.ac.uk,
h.alrashidi01@gmail.com

Abstract. Automatic analysis of reflective writing involves identifying indicator strings and using string matching or rule matching processes, which flag sections of a text containing reflective material. The problem with the string-based approach is its inability to deal with knowledge inference from the text, such as the content, context, relevance, clarity, and interconnection, which can be identified by semantic analysis. The semantic analysis depends mainly on mapping the text into stored knowledge sources, such as WordNet, and analyzing the associations in the underlying knowledge source. In this paper, a semantic-based approach for reflective writing analysis is proposed, in which the input text, which is being analyzed, is mapped into semantic concepts. Moreover, a machine learning (ML) approach for reflective writing identification and analysis has been implemented to overcome the limitations of rule execution and keyword matching. The proposed approach addresses the efficiency of using several effective concepts, correlated with effective words that are identified in WordNet-Affect. The input text is classified into reflective or non-reflective categories, after which the input text is classified into various reflective classes, based on the type of the document. Moreover, the concepts in WordNet-Affect are evaluated and analyzed to demonstrate their effects on classification and labeling tasks.

Keywords: WordNet-Affect · Classification · Automatic · Reflective · Semantic-based

1 Introduction

Reflective Writing (RW) involves insights and mental considerations of learned topics, past experiences, and actions. RW has several definitions in the literature, among them, being that it is a form of conceptual processing with a purpose that is applied to unstructured ideas in a case of solution [1], a purposeful philosophy toward a goal [2] and an efficient method of thinking about practice [3].

The benefits of RW are significant [4] since the potential of learning could be lost or forgotten without reflecting on the experience. Vass and Littleton [5], Chen, Wei [6], and Xie, Ke [7] found that the most important role of reflection in students' writing is to enhance their practice [8].

Y. Bi et al. (Eds.): IntelliSys 2019, AISC 1038, pp. 321–333, 2020.
https://doi.org/10.1007/978-3-030-29513-4_23

However, the benefits of RW come with difficulties in terms of RW analysis and feedback. RW analysis involves classifying an input text into either reflective or non-reflective components and providing feedback to enhance the reflecting writing by highlighting the strengths and weakness of the input text with regards to the properties of RW. Theoretical models of RW analysis have been proposed that involve using string indicators to determine whether the reflection is present [9]. Moreover, some models distinguish between different levels of reflection [10]. As the manual analysis is tedious and time-consuming, there is a need for automatic RW analysis [8].

To implement automatic reflective writing analysis as an analog to the existing in manual models, Natural Language Process (NLP) is used. NLP is a "potential for computational analysis of reflective writing (Reflective Writing Analytics) as a means of discovering evidence for metacognitive activity in the reflective writing of a learner" [11]. NLP has the potential to automatically analyze the input text and discover the reflection indicators to make a final decision about the input and to determine whether the text is reflective or not.

The existing approaches to automatic RW analysis, which are dictionary-based or rule-based, depend on shallow text processing and the extraction of string patterns to be used as input for rule evaluation and keyword matching.

The problems of existing automatic reflection analysis include the inability to deal with the depth of the reflection activities in the text [12]. This is because the existing approaches depend on the strings alone and ignore the semantic features of the text. While reflection depends on a deep understanding of the underlying topic and event, the automatic process does not give any attention to the depth of the content in the analyzed text.

In this paper, a semantic-based approach for RW analysis is proposed. The input text being analyzed is mapped into semantic concepts, and based on the efficiency of using several effective concepts is correlated with affective words. These are identified in WordNet-Affect which is a linguistic resource for a lexical representation of affective knowledge created by Strapparava and Valitutti [13].

1.1 Research Question

The goal of this paper is to distinguish between three levels for reflective text analysis automatically depends on the ML. To answer this research question;

How to automatically distinguish between descriptive, reflective, and critical reflective texts?

1.2 Contribution

This paper has two contributions: (1) the automatic RW analysis designed based on reflection theory, and (2) the automatic RW analysis based on semantic concepts concerning WordNet and WordNet-Affect and using the ML classification algorithms.

2 Related Work

Various theoretical models for RW analysis have been proposed, each of which classifies text into one of several categories that are defined precisely, in order to help an assessor to provide correct analysis and sufficient feedback to the writer. Automatic RW analysis has been built on top of these models, in order to ease the analysis task. Existing approaches for automatic reflection analysis are classified into keyword-based and rule-based categories. The keyword-based category depends on locating specific keywords, as an indication of reflection, in the input text, using a keyword matching process. The rule-based category depends on applying specific rules in sentences or phrases in the text.

2.1 Reflective Writing Models

Besides the different viewpoints of the researchers in this field, the variety of reflection models can be referred back to areas, types, and fields of reflection. Kember [14] categorizes reflective text into seven categories these are: (1) Habitual action, (2) Introspection, (3) Thoughtful action, (4) Content reflection, (5) Process reflection, (6) Content and process reflection, and (7) Premise reflection. While the first three are non-reflective, four to six are reflective, and the last is critically reflective. Plack et al. [15] categorize reflective text into three categories, namely non-reflective, reflective, and critically reflective. Accordingly, without losing any generality, this model can be used effectively as the basis for the automatic analysis of various fields and text types.

2.2 Keyword-Based Automatic Analysis

Ullmann [16] used various NLP tools to detect different reflection indicators. The model was trained using a dataset of labeled reflective texts to capture associations of different words with different reflection categories. This helps to constrain the consistency of feedback but fails to build a comprehensive model as it is based on string features of the text. An enhanced bag-of-words model for automatic RW assessment was proposed by El-Din [17], which takes into account a sentiment scores at the word level to improve feedback.

However, the sentiment is just one of the feedback attributes. Gibson et al. [11] proposed a more comprehensive keyword-based approach which categorizes text based on several metacognitive activities. The aim is to examine the extent to which the conceptual model may correspond to lexical and structural features in RW. The conceptual model includes RW features that were identified from previous studies, some of these being pronouns, adapted from the work of Pennebaker and Chung [18], and linguistic features adapted from Ryan [19].

2.3 Rule-Based Automatic Analysis

Academic writing analytics have been proposed by Shum et al. [20] to provide an educational interface, where a single NLP tool, Xerox Incremental Parser (XIP) [21], is used to detect and label reflective sentences without evaluating the whole document as reflective or not. XIP implements syntactic analysis, lexical resources, and the dependency rules that detect the reflective patterns.

A prediction model was developed and implemented by Chen et al. [22] to discover a common topic discussed in students' RW. The strength of this model is that it uses various classifiers to categorize the strengths and weaknesses in the RW, such as Naive Bayes, Decision Tree J48, and Support Vector Machines.

However, while this model helps to answer some of the issues faced, it fails to give a *comprehensive* solution. As noted, these studies demonstrate the ability to conduct automatic RW assessment using NLP tools, techniques, and rules. One of the interesting points about these studies is the investigation of the string features of the text regardless of the semantic features.

3 Proposed Work

The proposed approach for reflective text analysis depends on ML to implement coarse- and fine-classification tasks. In the coarse-classification task, the input text is classified into reflective or non-reflective categories, while in the fine-classification task, the reflective text is labeled with one or more of the thinking activities that can be related to the reflective activities.

The proposed approach depends on extracting and using a set of features as input to a classification algorithm, so as to generate a specific class or label to the input text. The features correspond to semantic concepts related to WordNet conceptual knowledge source, and these are emotion, mood, trait, cognitive state, physical state, hedonic signal, emotion-eliciting situation, emotional response, behavior, attitude, and sensation. The implemented approach is illustrated in Fig. 1.

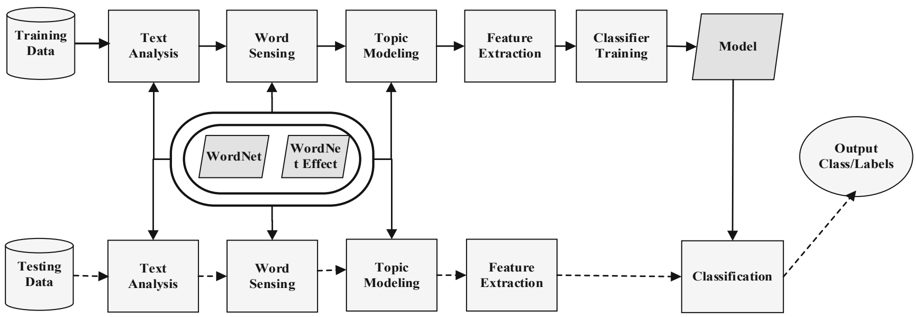

Fig. 1. RW analysis approach

First, the text is analyzed and mapped to its corresponding concept in WordNet-Affect. Second, the word sense is determined using word sense disambiguation. Then, the topic of the text is modeled with reference to WordNet-Affect. A feature vector of WordNet-Affect concepts is constructed, and finally, the feature vector is used to classify the input text, using classification algorithms, into reflective or non-reflective categories. After this, the input text is classified into various activity classes as identified by Plack et al. [15], namely non-reflective, reflective, and critically reflective.

3.1 Text Analysis

In text analysis, the input text is tokenized and tagged with its part-of-speech to be lemmatized. The lemmas of the input words are extracted using Stanford Tagger [23]. It is important to identify the parts of speech and the lemmas of each word to map the word into its associated WordNet synsets, which form a lexical database of English. Nouns, verbs, adjectives, and adverbs are grouped into sets of synonyms called *synsets* because WordNet presents lexical knowledge-based and arranges words of similar meaning into synsets. Relations and associations between the components of the knowledge base are constructed based on the synsets, not the words [24]. The steps implemented by the text analysis stage are illustrated in Fig. 2.

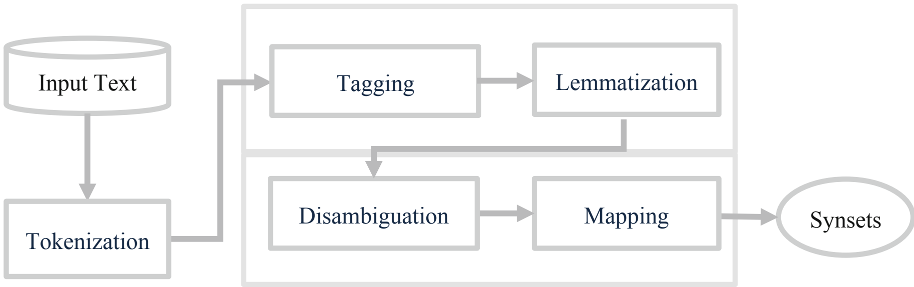

Fig. 2. Text analysis stage

3.2 Word Sense Disambiguation

In mapping words to WordNet, the same word might exist in a different synset based on its part of speech and its lemmas. Moreover, the same word might exist in various synsets depending on its meaning. Subsequently, the input text is then disambiguated to map each word onto the correct WordNet synsets. The disambiguation process is implemented based on the semantic similarity among the words in the text. Various semantic similarity measures exist, which can be designated as information-based and relation-based.

In this paper, a word is disambiguated by choosing the synset of the word that most overlaps the synsets of other words in the sentence using Lesk [25]. As given in Eq. 1. If more than one synset obtains the same overlapping rate, according to the Lesk measure, then, the most frequent meaning is selected, where, $comp_i$ and $comp_j$, are the components of the word of the underlying sense (e.g.: synset) and the component of a given the word in the sentence, $x_{i,\,j}$ is equal to one if the components are identical and zero otherwise The output value of Eq. 1 is divided by the length of the gloss to be normalized in the range [0–1].

$$Overlap(sense, word) = avgmax \sum_{i,j} \left(x_{i,j} f(comp_n == comp_n \right) \tag{1}$$

Lesk is used as a similarity measure, where the similarity between a pair of words is calculated as the number of common words number in their definitions, which is called *gloss* in WordNet. To extend this idea, in this paper, each possible synset is given an overlapping rate value that represents its maximum similarity with any synset of any word in the input sentence. The output synset of each word is selected as the one with the highest overlap among them all. In doing so, the disambiguation process is implemented as collectively determining the synsets of all words in the sentence.

3.3 Topic Modeling

In this step, the input synsets are modeled using WordNet-Affect (see Fig. 3) [13]. However, instead of using the direct mapping into WordNet-Affect, these synsets are used to model the topic of the text, as reflective analysis depends on sentences rather than the whole text and short sentence classifications cannot be implemented directly.

Accordingly, each sentence is linked to all WordNet-Affect categories. As the synsets for the input text are extracted, the similarity of each synset with each category in WordNet-Affect is calculated using the Lesk measure. Accordingly, each extracted synset in the input text will have a vector of values the length of which is equal to the number of categories in WordNet. Given this, there are multiple synsets for each category in WordNet-Affect, and the similarity is calculated as the maximum similarity of any of these synsets, keen on the same concepts of using Lesk for disambiguation, in the previous stage. At the end of this stage, the input text is represented by WordNet-Affect labels instead of words.

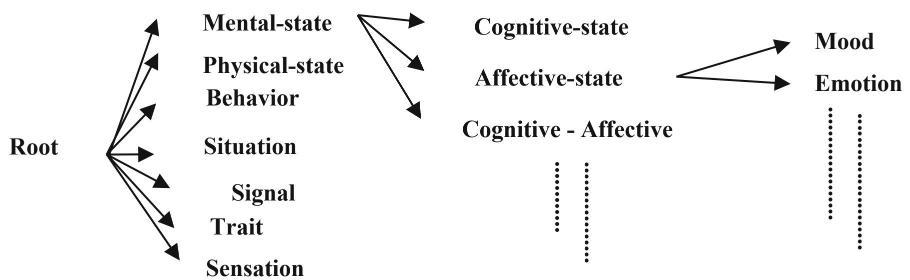

Fig. 3. WordNet-Affect hierarchy [13]

3.4 Feature Vector Construction

For each input text, a feature vector of 310 elements is created. The feature vector constructed in this process corresponds to the number of concepts in WordNet-Affect, which is 309, while the last element in the vector corresponds to the class value (non-reflective/reflective/critically reflective). The feature vector, of each input text is filled by the maximum values for its synsets. To avoid underfitting, as the number of features, are huge, feature selection is implemented after the feature vectors of all input texts are extracted.

3.5 Classification

The feature vectors of RW texts and non-RW texts are fed into the classifier, and a model is trained to be used in the prediction phase. The classification algorithms can be broadly classified into a decision tree based algorithms, probability-based algorithms, and instance based algorithms and support vector machines.

A classification algorithm for each of these categories is utilized in the proposed work. The decision tree classification [26] algorithms construct a model in the form of a tree, in which the internal nodes denote a single feature in the feature vector, the branches going out from each internal node represents the values of that feature, and the leaf nodes represent class labels.

In the model construction, the training examples are used recursively, based on the values of their features to construct the best tree that can fit with these examples. Similar models are constructed in the support vector machine and the probability-based algorithms. An instance-based model does not construct a trained model and uses the instances in the training phase to predict the class of given samples in the testing phase. Among these categories, using a decision tree-based classification will allow a determination of the influence of each feature on the task of RW detection. Moreover, the decision tree classifier can be implemented easily [27–29].

3.6 Experiments and Results Analysis

To evaluate the proposed approach for RW detection, a dataset, which is formulated for the evaluation, and a program for processing text and analyzing the generated results, are presented. After obtaining the results, the efficiency of using WordNet-Affect is evaluated accordingly.

4 Experiments and Results

The first step in the experiments was to give a precise definition to each category in the model. Accordingly, Table 1 summarizes the definition of the model and maps it to the Description.

Table 1. Reflective classes description adopted from Plack et al. model [15]

Class	Description
Non-Reflective	The writer attempts to describe the fact of the experience rather than analyzing the experience, related literature, existing techniques, theories, other concepts
Reflective	The writer attempts to deconstruct the investigation experience, analyze evidence, and differentiate/contrast results and causes
Critically Reflective	The writer attempts to draw conclusions, propose suggestions and/or new ideas, and evaluate alternative solutions

4.1 Dataset

Overall, there is a lack of RW corpora to support RW research. For the evaluation purposes, British Academic Writing English Corpus (BAWE), is used [30]. The corpus includes a set of student writings in various fields of study, including Architecture, Chemistry, and Computer Science. Each assignment is graded with M (Merit) and D (Distinction). The corpus involved 13 different assignment formats, including case study, critique, and literature survey.

This dataset was not created for reflection studies and does not classify the type of text into reflective/non-reflective. Subsequently, in this paper, pre-analysis of this corpus was conducted in order to use this corpus in the proposed approach. First, a single file from each assignment format was selected, from various fields of study. Second, only assignments with distinction mark (D) were considered in the experiments to ensure that the involved text met the description given by the corpus. The total number of sentences were used in the experiments was 979. The constructed dataset summary is given in Table 2.

Table 2. Description of the constructed dataset

Format	Field	Language	Mark	No. of sentences
Case Study	Engineering	English	D	42
Critique	Computer Science	English	D	105
Design Specification	Computer Science	English	D	115
Empathy Writing	Engineering	English	D	39
Essay	Economics	English	D	100
Exercise	Computer Science	English	D	64
Explanation	Engineering	English	D	49
Literature Survey	Philosophy	English	D	20
Methodology Recount	Engineering	English	D	31
Narrative Recount	Engineering	English	D	44
Problem Question	Engineering	English	D	114
Proposal	Engineering	English	D	140
Research Report	Economics	English	D	116

The pre-analysis of this corpus is as follows: the text was normalized to remove any markup, then sentences from each document were extracted. The sentences were annotated manually by experts with one of the following classes: non-reflective, reflective and critically reflective, and examples are given in Table 3. The numbers of sentences in each category were as follows: non-reflective had 529 sentences, reflective had 427 sentences and critical reflective had 23 sentences based on Plack et al. model [16], see Table 1.

Table 3. Example sentences in the dataset

Sentence	Category
"I read quite a wealth of journals and papers before embarking, so the content just flowed when actually writing the essay."	Non-Reflective
"I think the thing I found most difficult was remembering to use the right brackets all the time in the functions, and remembering which variables contained lists, and which contained single variables."	Reflective
"I now better understand how focusing questions maximises search results (Thompson and Dowding 2002) as I found that I needed to refine my search as I went along, so having a better idea of how to find relevant information would have improved my searching and made better use of time; however I do feel I was able to access adequate and appropriate information."	Critically Reflective

4.2 Evaluation Measures

The output of the detection and classification tasks are evaluated using classification accuracy, precision, recall, and the F-measure. These measurements are based on the proportions, which are True Positive, True Negative, False Positive and False Negative, and relate the ground truth and the predicted solution by the assessed method, as follows. True Positive is the size of the correctly detected portion, False Positive is the size of the wrongly detected portion, True Negative is the size of the correctly unpredicted portion, and False Negative is the size of the wrongly unpredicted portion.

Accuracy is the ratio between True Positive and True Negative to the number of overall items. Precision is the ratio between True Positive and the overall number of predicted items by the underlying approach or method.

The overall predicted portion combines both the True Positive and the False Positive. The recall is the ratio of the True Positive to the true solution represented by the True Positive and the False Negative. F-measure combines precision and recall as a mean of a single indication of efficiency.

4.3 Experimental Design

The design of the experiments was conducted using the Java programming language and a set of libraries and resources. The input text is tokenized, and all non-word tokens are removed. Tokenization is implemented using the delimiter list, which involves the set of delimiters commonly used in the tokenization process. Next, the regular expressions are used to distinguish letters from non-letters and remove non-words accordingly.

Stanford Tagger is used to determine the part of speech for each word and its correct lemma. JWNL (Java WordNet Library) [31] is used to map the words into their association synsets in WordNet. The synsets are then labeled based on WordNet-Affect labels (version 1.1).

The results of these process are vectorized and arranged in ARFF (Attribute-Relation File Format) to be used with the WEKA Tool and Library [32], which implements various classification algorithms. As mentioned before, several forms of the feature vector are generated and will be used in the classification process. The classification is conducted using a 10-fold process, where in each fold the input samples are divided into training and testing sets.

4.4 Result Analysis

Experiments with four classification algorithms were conducted, and the accuracy of the outputs are given in Fig. 4. As noted, in the first task, in which only reflective vs. non-reflective classification is considered, SVM gives the best accuracy that slightly outperforms Bayesian, which falls in the second rank.

Dividing the reflective data into reflective and critical reflective categories in the second task, reduces the results slightly for SVM and Bayesian, and surprisingly enhances the accuracy of the rest. Subsequently, using SVM for detection and classification of reflective text, obtained the best accuracy.

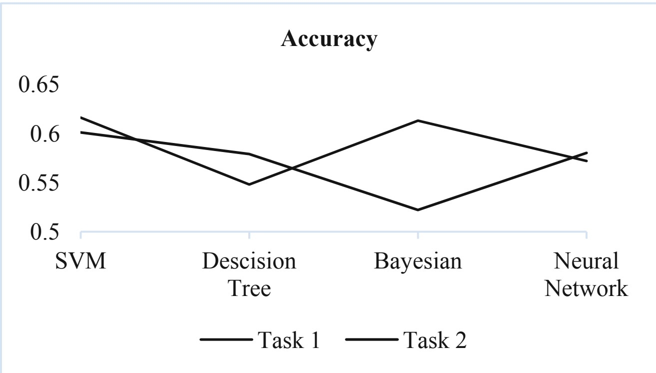

Fig. 4. Accuracy comparison between classification algorithms

The discriminating ability of the WordNet-Affect concepts are examined in two ways: the first one is by drawing part of the tree that is constructed in the learning phase of the decision tree classification. The presence of most of the concepts (the edges with values greater than zero), such as "comfortableness," indicates, mostly, the presence of reflection (the class with the value of one in the rectangle) and vice-versa. Second, a randomly selected feature selection algorithm, which is the best first algorithm that selects a sub-set feature of the original set, was executed. The output of feature selection was a sub-set of 19 concepts out of the 309, as given in Table 4. The classification task using these compact set was re-implemented.

Table 4. Compact subset of WordNet Affect concepts

Levity	Comfortableness	Gloat	Soft-spot
Negative-fear	Humility	Apprehension	Timidity
Shyness	Downheartedness	Despondency	Annoyance
Aggravation	Misogyny	Discomfiture	Hopelessness
Pessimism	Ambiguous-agitation	Buck-fever	

The proposed approach for detecting reflection writing text is compared with the string-based approach, based on a set of keywords that are listed and experimented by Ullmann [12]. The results of the proposed approach and the string-based approach are summarized in Table 5. The results of the proposed approach slightly outperform the results of the string-based approach. Besides the accuracy, as mentioned, the proposed approach can be extended to analyze the content as it is based on semantic concepts.

Table 5. Reflective text detection results comparison

	The proposed approach	String-based
Accuracy	0.616	0.606
Precision	0.627	0.547
Recall	0.715	0.846
F-Measure	0.668	0.664

5 Limitation

There are some limitations faced during this paper; one particular challenge is applying NLP approaches. In order to understand the meaning of natural language; machines have to learn how to do machine learning (ML) within NLP. Because the ML algorithm is a wide range of basic complex algorithms. Which apply these to a specific domain is a relative exercise, are integrated with a text-based approach to reach the target of this work. Human ability outperforms the automatic approaches regarding reliability and accuracy of RW detection. The boundaries are used to convert sentences into features that are required to train the ML approach.

6 Conclusion

Automatic reflective analysis based on semantic concepts concerning WordNet and WordNet-Affect and using ML classification algorithms is conducted in this paper. The use of semantic features forms the core of semantic and intelligent RW analysis and presents an in-depth analysis of the reflection activities in the text. As such, while these semantic features are used, in this paper for the detection task, it can be further utilized in evaluation, categorization, or advancing the RW tasks.

Semantics are obtained in the proposed approach by mapping words into WordNet concepts and then locating these concepts in WordNet-Affect. These concepts are then used to create a feature, which is forwarded as input to the classification algorithm that labels the text with a reflective or non-reflective label.

In order to evaluate the proposed approach, we use the British Academic Writing English Corpus (BAWE), which includes a set of student writings in various fields of study and various assignment formats, including case study, critique, and literature survey. However, a pre-analysis for this corpus was required in order to fit with the task at hand. The results showed that WordNet-Affect for RW detection was sufficient.

The analysis of the examined discriminating ability of the WordNet Affect concepts showed that the presence of most of the concepts, such as "ambiguous" or "negative-fear," indicates, mostly, the presence of reflection and vice-versa. However, a subset of 19 concepts out of the 309 was sufficient to detect reflection with mostly identical accuracy, rather than using the whole set of 309 concepts.

The results showed that the pro-posed approach outperformed the string-based approach. This indicated that classification reflection depends on the probability of presence/absence of some concepts in WordNet-affect that comes in line with the current non-semantic RW detection, and which depends on presence/absent of exact words. However, this is unlike non-semantic RW detection, which depends on locating words.

Acknowledgment. We would like to acknowledge the contribution of this research, that is funded by Kuwait Foundation for the Advancement of Sciences (KFAS) under project code CB19-68SM-01, which has part-funded this research.

References

1. Moon, J.: Learning Journals. A Handbook for Academics, Students and Professional (1999)
2. Dewey, J.: A restatement of the relation of reflective thinking to the educative process. DC Heath (1933)
3. Calderhead, J.: Teachers: beliefs and knowledge. In: Berliner, D.C., Calfee, R.C. (eds.). Macmillan, New York (1996)
4. Thorpe, K.: Reflective learning journals: from concept to practice. Reflective Pract. **5**, 327–343 (2004)
5. Vass, E., Littleton, K., Miell, D., Jones, A.: The discourse of collaborative creative writing: peer collaboration as a context for mutual inspiration. Think. Ski. Creat. **3**, 192–202 (2008)
6. Chen, N.-S., Wei, C.-W., Wu, K.-T., Uden, L.: Effects of high level prompts and peer assessment on online learners' reflection levels. Comput. Educ. **52**, 283–291 (2009)
7. Xie, Y., Ke, F., Sharma, P.: The effect of peer feedback for blogging on college students' reflective learning processes. Internet High. Educ. **11**, 18–25 (2008)
8. Moseley, D., Baumfield, V., Elliott, J., Gregson, S., Higgins, S., Lin, M., Miller, J., Newton, D., Robson, S.: Thinking skills frameworks for post-16 learners: an evaluation (2004)
9. Wong, F., Kember, D., Chung, L.: Assessing the level of student reflection from reflective journals. J. Adv. Nurs. 22: 48-57 (1995)
10. Sumsion, J., Fleet, A.: Reflection: can we assess it? Should we assess it? Assess. Eval. High. Educ. **21**, 121–130 (1996)

11. Gibson, A., Kitto, K., Bruza, P.: Towards the discovery of learner metacognition from reflective writing. J. Learn. Anal. **3**, 22–36 (2016)
12. Ullmann, T.D.: Keywords of written reflection-a comparison between reflective and descriptive datasets. CEUR Workshop Proc. **1465**, 83–96 (2015)
13. Strapparava, C., Valitutti, A.: Wordnet affect: an affective extension of WordNet. In: LREC, Lisbon, p. 40 (2004)
14. Kember, D.: Determining the level of reflective thinking from students' written journals using a coding scheme based on the work of Mezirow. Int. J. Lifelong Educ. **18**, 18–30 (1999). https://doi.org/10.1080/026013799293928
15. Plack, M.M., Driscoll, M., Marquez, M., Cuppernull, L., Maring, J., Greenberg, L.: Assessing reflective writing on a pediatric clerkship by using a modified Bloom's taxonomy. Ambul. Pediatr. **7**, 285–291 (2007)
16. Ullmann, T.D.: Keywords of written reflection - a comparison between reflective and descriptive datasets. In: CEUR Workshop Proceedings, pp. 83–96 (2015)
17. El-din, D.M.: Enhancement bag-of-words model for solving the challenges of sentiment analysis, **7**, 244–252 (2016). https://doi.org/10.14569/IJACSA.2016.070134
18. Pennebaker, J.W., Chung, C.K.: Expressive writing: connections to physical and mental health. In: Friedman, H.S. (ed.) Oxford Library of Psychology. The Oxford Handbook of Health Psychology, pp. 417–437. Oxford University Press, New York (2011)
19. Ryan, M.: Improving reflective writing in higher education: a social semiotic perspective. Teach. High. Educ. **16**, 99–111 (2011). https://doi.org/10.1080/13562517.2010.507311
20. Shum, S.B., Sándor, Á., Goldsmith, R., Wang, X., Bass, R., McWilliams, M.: Reflecting on reflective writing analytics: assessment challenges and iterative evaluation of a prototype tool. In: Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, pp. 213–222. ACM (2016)
21. Aït-Mokhtar, S., Chanod, J.-P., Roux, C.: Robustness beyond shallowness: incremental deep parsing. Nat. Lang. Eng. **8**, 121–144 (2002)
22. Chen, Y., Yu, B., Zhang, X., Yu, Y.: Topic modeling for evaluating students' reflective writing. In: Proceedings of Sixth Interantional Conference on Learning Analytics & Knowledge - LAK 2016, pp. 1–5 (2016). https://doi.org/10.1145/2883851.2883951
23. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 173–180. Association for computational Linguistics (2003)
24. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**, 39–41 (1995)
25. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation, pp. 24–26. Citeseer (1986)
26. Murthy, S.K.: Automatic construction of decision trees from data: a multi-disciplinary survey. Data Min. Knowl. Discov. **2**, 345–389 (1998)
27. Beattie, D.: Experiments in Induction. In: Hunt, E.B., Marin, J., Stone, P.J. (eds.) Academic Press, New York (1966). xi+247 pp. $9.50 (1969)
28. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees (Chapman y Hall, eds.) Monterey, CA, EE. UU. Wadsworth Int. Gr. (1984)
29. Kononenko, I.: Estimating attributes: analysis and extensions of RELIEF. In: European Conference on Machine Learning, pp. 171–182. Springer, Heidelberg (1994)
30. Heuboeck, A., Holmes, J., Nesi, H.: The BAWE Corpus Manual (Version III). Publ. por Univ. Coventry (2010)
31. Walenz, B., Didion, J.: JWNL: Java WordNet Library (2011)
32. Holmes, G., Donkin, A., Witten, I.H.: Weka: A Machine Learning Workbench (1994)