Lexicon-Based Sentiment Analysis of Student's Feedback for Teacher's Evaluation

Francis B. Oliveira[1]

[1]Center of Mathematics, Computation and Cognition - CMCC
Federal University of ABC - UFABC

francis.oliveira@ufabc.edu.br

Abstract. *This paper describes my efforts to implement, get similar results and improvements over the ones achieved by the authors of article [Rajput, Haider and Ghani 2016]. I followed the same techniques used in the original article, but writing all code from scratch rather than using a tool to auto-generate. Course evaluation has become an integral part of education management in almost every academic institution. The existing evaluation method primarily employs the Likert scale based quantitative scores and open-ended questions where students write general comments/feedback. The textual feedback, however, is usually provided to teachers and administration and due to its non quantitative nature is frequently not processed to gain more insight. This article aims to address this aspect by applying several text analytics methods on student's feedback and also provides a new insight into a teacher's performance with the help of tag clouds and sentiment score.*

1. Introduction

Evaluating performance of teachers is very important for all academic institutions, from student's feedbacks we can extract some insights that can help to improve the quality and content of the courses.

The evolution method used in the original article is based on a form that is typically collected at the end of each course on a set of questions which are answered using Likert scale. The aggregate of the responses is used as a metric to gauge the teaching quality of the concerned faculty member.

The evaluation form, however, also provides room for open feedback which typically is not included in the performance evaluation/appraisal due to lack of automated text analytics methods [MacKim and Calvo 2010], [Jagtap and Dhotre 2014]. The textual data may contain useful insight about subject knowledge of the teacher, regularity, and presentation skills and may also provide suggestions to improve the teaching of quality. Such information may not be readily extracted from the Likert scale based feedback [Ogden and Lo 2010]. However, getting sense out of the textual feedback manually is a laborious task and, as a result, the textual feedback is not properly utilized [Jagtap and Dhotre 2014].

The primary focus of the sentiment analysis is to determine a writer's feeling from a given text. The feeling might be his/her attitude, emotion, or opinion. The most important step of this analysis is to classify the polarity of the given text as positive, negative, or neutral [Medhat, Hassan and Korashy 2014]. In a similar fashion, the presented work aims to identify the polarity of a student's feedback in terms of positive, neutral, and negative. In addition, the article also suggests methods to identify the recurring theme in students' feedback by generating word clouds for visualization and sentiment score.

This article reproduces the results obtained in [Rajput, Haider and Ghani 2016], and additionally apply an algorithm called *Vote & Flip [*Choi and Cardie, 2009*]* to improve the accuracy of sentiment score results. Besides the original article uses Likert scale it didn't provide any data regarding feedbacks using this technique, and to avoid assumptions we'll not explore it in this article.

The rest of the article is organized as follows: Section 2 provides a brief survey of the field of sentiment analysis. Section 3 discusses the presented approach for analyzing students' feedback while the results are presented in Section 4 with a comparison between the original and the adapted implementation proposed in this article. Finally, Section 5 concludes the article and provides future research directions.

2. Related Work

Sentiment Analysis is a field that has been getting more attention in the last years and many companies, from different business areas, demonstrated interest in understand customers thoughts and feelings about their products and services by automatically analysing customer's feedback. Researchers developed some techniques for automatic extraction of opinion or sentiment from text:

Keyword spotting: Keyword spotting is the most naive approach, text is classified into affect categories based on the presence of fairly ambiguous affect words like "happy", "sad", "afraid" and "bored". The weakness of this approach lie in two areas: poor recognition of affect when negation is involved and reliance on surface features.

Lexicon-Based Approaches: [Kim and Hovy 2004] proposed a dictionary based approach, that consists of a small set of opinion words collected manually as a seed, and then a dictionary or thesaurus is used to expand the set of opinion words by adding their synonyms and antonyms, the newly words are added to the seed. The major drawback of this approach is that is unable to find a domain and context specific opinion words.

The corpus based approach overcome the limitations of the dictionary based approach, in addition to seed word list, this approach identifies context specific opinion words. The finding of such words is based on syntactic or cooccurrence pattern in the text using linguistic constraints.

Lexical Affinity: This approach is based on the probability training from a linguistic corpora [Medhat, Hassan and Korashy 2014]. It only detects words without consider the phrase context, for example: In the phrase "He died by a car accident", the word "accident" has 75 percent probability to be a negative word, just as the same word in the phrase "I got promoted by accident".

Statistical Methods: This method uses machine learning algorithms like Naive Bayes or SVM to train models to discover polarity of the phrases. They generally use a labeled dataset from a specific topic, like customer product reviews for example, and the system learns the shades of affect words, punctuation and word cooccurrence frequencies. To be effective, this approach depends upon the quality and quantity of the training data and feature selection [Jagtap, Dhotre 2014].

Concept Level Techniques: Concept level techniques are leveraged on elements from knowledge representation such as ontologies and semantic networks and, hence, can detect semantics that are expressed in a subtle manner[Cambria, 2013].

In all the above approaches, the fundamental step of the sentiment analysis is to identify the polarity as positive, negative, or neutral on either a word level, sentence level, or document level. To facilitate this step, various subjectivity corpora are developed that annotate the lexicon of words with positive, negative, and neutral polarity. In applications where a domain is unknown, the use of such corpus may work fine. However, it is argued that better results can be achieved using a domain specific sentiment language. For this reason, several sentiment annotated corpora have been made freely available. The list includes the MPQA newswire corpus [Wiebe, Wilson and Cardie 2005] and restaurant and laptop review corpus [Medhat, Hassan and Korashy 2014]. It must be noted that, as of now, no such sentiment corpus is available exclusively for the educational domain.

Sentiment analysis can be applied to a broad range of real world problems. Many businesses are adopting text and sentiment analysis and incorporating it into their processes.

In the medical domain, physicians and nurses express their judgments and observations on a patient's health status in clinical narratives. Sentiments in clinical documents differ from the sentiments in user generated content or other text types. Analyzing and aggregating this information over time can support the treatment decisions by allowing a physician to quickly get the health status overview of a patient. [Goeuriot, Kelly and Zobel 2014] performed dictionary based sentiment analysis on clinical text.

In the education domain [MacKim and Calvo 2010] analyze student's feedback to evaluate their learning experience. The study compares categorical model and dimensional model making use of five emotion categories: anger, fear, joy, sadness, and surprise.

[Leong, Lee and Mak 2012] applied sentiment analysis and text mining by collecting student's feedback that is collected through SMS. Each feedback has been categorized based the concepts defined for each category.

[Jagtap and Dhotre 2014] used SVM and HMM based hybrid approach for sentiment analysis of teachers' assessment.

[Altrabsheh, Gaber and Cocea 2013] analyzed student's feedback by collecting via social media such as Twitter. They not only identified student's feeling in terms of positive and negative but also identified some more refined emotions like confused, bored, irritated, confident and enthusiastic. Different techniques have been used in sentiment analysis, and a few have proved to give superior performance such as Naive Bayes (NB), Max Entropy (MaxEnt), and Support Vector Machines (SVM).

3. Methodology proposal

This section explains the process used to analyse the student's feedback and identify the polarity of a comment as positive, negative or neutral. It's basically used the same process described in the original article, with the addition of vote & flip algorithm, that is the change proposed here.

Table 1 shows some student's feedback, the comments are short, some use abbreviated shortcuts and an informal writing style. These issues highlight the difficult in accurately extracting the sentiments from unstructured data.

Due to the unstructured nature of the comments, we need to use some techniques to process data and generate meaningful insights from it. Figure 1 illustrates the steps used to process the data and we will describe each step in the following sections.

Table 1. Samples of student's textual feedback

1	A good teacher
2	I felt that the instructor could've done a better job in making the concepts clear
3	Dull course
4	A course is fine
5	Difficult course, great teacher and also able to relate it to practical knowledge
6	This course increased my knowledge. Teacher is also very helpful I like this course very very much
7	Institute should reconsider its policy of fyp's

3.1. Preprocessing

The aim of preprocessing is to remove the unwanted and noisy data. In this paper, the preprocessing stage comprises the following tasks:

Tokenization: This process breaks a stream of text into a list of words.

Stemming: This step reduces words to their stem or root form as stemming simplifies the sentiment analysis process. The same word can be used in a different flavour for grammatical reasons such as organize, organizes, organizing.

Case conversion: This step changes the text into either the lowercase or the uppercase.

Punctuation Removal: Punctuations in a text generally do not provide any useful information. This step, therefore, erases the punctuation characters from the word.

Stop Word Removal: Stop words consist of prepositions, help verbs, articles, and so forth. They typically do not contribute in analysing sentiments and are removed from the text.

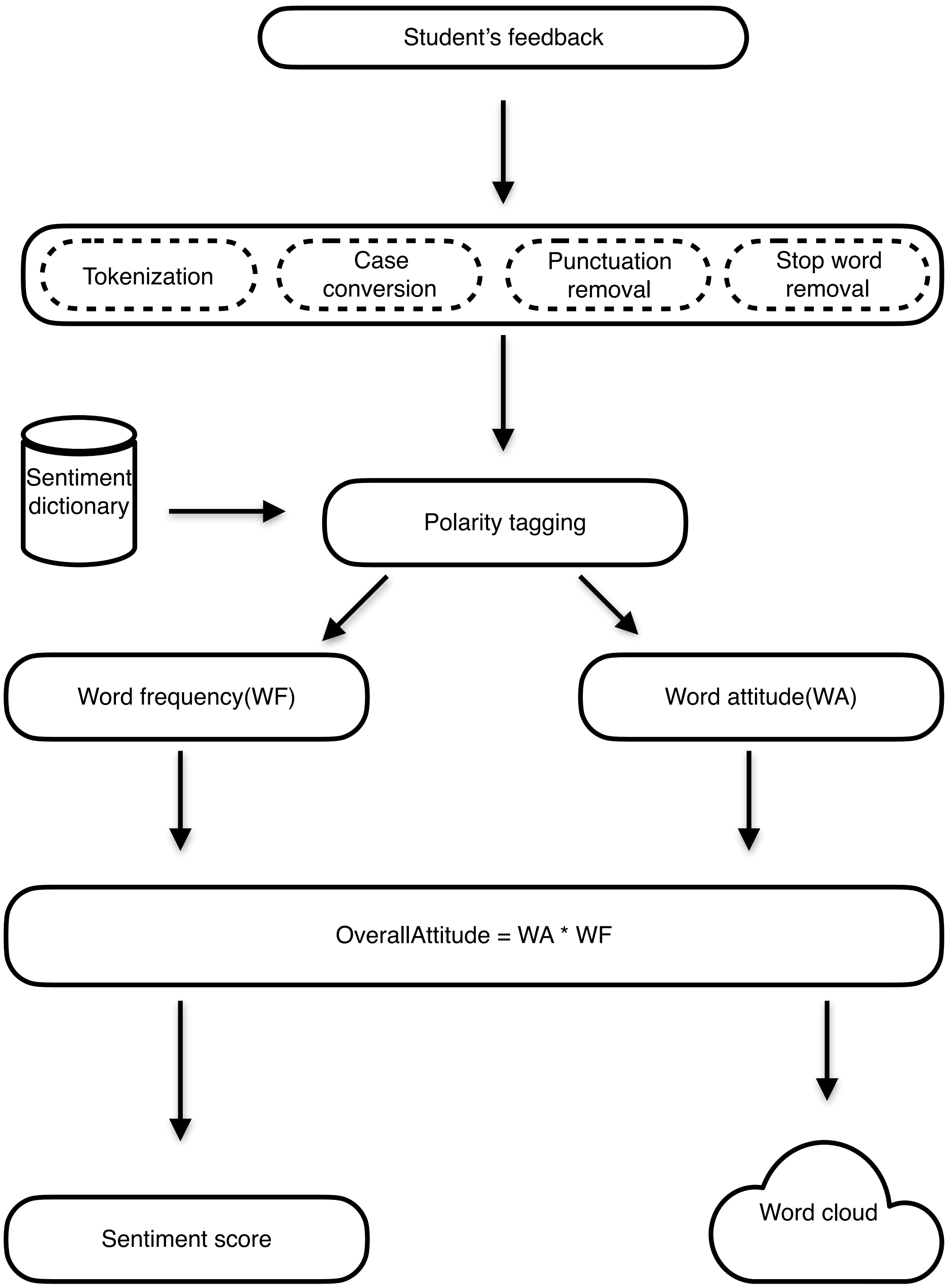

Figure 1. Sentiment analysis process

3.2. Sentiment Dictionary

A sentiment dictionary contains a list of words along with their respective polarity. Several such corpora have been developed and are made freely available. In this work, we have used the MPQA corpus [Wiebe, Wilson and Cardie 2005].

3.3. Polarity Tagging

This step analyses each word in a student's feedback and tags the word as positive, negative, and neutral using its polarity in the sentiment dictionary. The neutral words are removed from the data as they do not provide any subjectivity clue. For example, consider a student's feedback as shown in row 5 of Table 1:

"Difficult course, great teacher and also able to relate it to the practical knowledge."

In this example, after the preprocessing step, each word would have been tagged with their polarity as found in the sentiment dictionary. Here the word difficult is tagged with negative polarity while great, able, and practical are tagged with positive polarity as shown in the Table 2:

Table 2. Polarity words tagged from student feedback

Sentiment words	difficult	great	able	practical
Word polarity	negative	positive	positive	positive

3.4. Word Frequency

This step computes the frequency of each word in each comment. In the example of table 3 taken from feedback of previous section, each word occurs only one time; therefore frequency of each word is one:

Table 3. Words frequency from student feedback

Sentiment words	difficult	great	able	practical
Word frequency	1	1	1	1

3.4. Word Attitude

This step converts the polarity of each word into a numeric value to perform further computation. If polarity is positive, the associated value is 1, in case of negative the value is -1. See the example in Table 4:

Table 4. Words attitude from student feedback

Sentiment words	difficult	great	able	practical
Word attitude	-1	1	1	1

3.5. Overall Attitude

The overall attitude of a word is obtained by multiplying its attitude with its frequency:

$$OverallAttitude = Word\ Attitude * Word\ Frequency$$

In the example above, since the frequency of each sentiment word is 1 the overall attitude of these words is is either -1 or +1.

3.5. Word Cloud Visualisation

The overall attitude of each sentiment word from all feedbacks, will be used to draw a word cloud.

3.6. Sentiment Score

In this step, each feedback comment is assigned a sentiment score by adding the overallAttitude of each word in a feedback. This score is then used to evaluate a teacher's performance:

$$\text{SentimentScore} = \sum_{i=1}^{n} overallAttitude(i)$$

N is the number of positive and negative words in a feedback and *i* represents a particular word. The sentiment score is computed by adding overallAttitude of all positive and negative words while ignoring the neutrals. The sentiment score is the summation of overallAttitude of each sentiment word in a feedback. For the example used in previous sections, the sentiment score is:

$$\text{SentimentScore} = (-1) + 1 + 1 + 1 = 2$$

3.7. Vote & Flip

Some feedbacks have words that change the context of a phrase but can't be identified in the original implementation, for example:

The teacher is not good

In the example above, this feedback would be classified as positive, running the algorithm from original implementation, because it would just consider the word "good" in this phrase, but it's not the correct polarity. The word "not" changes the phrase context and it's clear that is a negative comment. To solve this problem we need to take into consideration negation words found in the comments and change the polarity according to the number of negation words found.

To (re)calculate the sentiment score of a phrase, according to negating words, we use a similar version of the Vote & Flip algorithm used in [Choi and Cardie, 2009] as you can see in figure 2.

4. Results

In this section we'll present the results achieved by the original article and compare with the results from the modified version using the *Vote & Flip* algorithm. The authors of the original article conducted an experiment analysing some feedbacks from more than 1700 students but they didn't release all the data, just some samples showed in the paper. To reproduce the results, I used all feedback samples published in the original article and added more feedbacks, relative similar, to have more data and do the comparison. In total, we'll use 51 feedbacks to produce the results.

```
For each feedback fi,
  priorPolarity = calculateSentimentScore
  nNegator = # of negating words in fi

  if (nNegator % 2 == 0)
      then flipPolarity = true
  else
      then flipPolarity = false

if (priorPolarity == positive) & (flipPolarity  == false)
    then finalPolarity(fi) = positive
else if (priorPolarity == positive) & (flipPolarity == true)
    then finalPolarity(fi) = negative
else if (priorPolarity == negative) & (flipPolarity == false)
    then finalPolarity(fi) = negative
else if (priorPolarity == negative) & (flipPolarity == true)
    then finalPolarity(fi) = neutral
else
    then finalPolarity(fi) = priorPolarity
```

Figure 2. Vote & Flip Algorithm

Another difference from original article is the way used to process the data. In the original article they used a tool called *Knime,* that is an open source platform for data analytics and allow users to create workflows and integrate various components for data mining and text processing. For this article I implemented all rules and algorithms writing all the code from scratch using the Java language and some open source libraries to do the *stemming* and generate the *word cloud*. Before show the results, we'll see the changes in dictionary made in original article to approximate the lexicon to the student's language.

4.1. Sentiment Dictionary Modification

Due to the informal way of writing from student's, some words may have a different meaning depending on the context. The word miss, for example, is assigned a negative polarity in the dictionary, but in the feedback bellow, it has a positive meaning:

Mis xyz is an awesome teacher

Table 5 shows a list of words that had it original polarity changed for this experiment.

4.2. Feedback analysis using Word Cloud

Word cloud visualisation is an excellent way to show the results. In the original article, they used *Knime built-in* word cloud generator. In this experiment we used a Java library that receives the input and generates a word cloud. Figure 3 shows an example with a word cloud generated with all student's feedback used in this experiment.

4.3. Feedback analysis using Sentiment Score

The original paper suggests the computation of a sentiment score for each feedback. A negative score means a negative comment, a positive score a positive and 0 indicates a mixed feedback(equal number of positive and negative words). Table 6 shows a comparison with the sentiment score calculated for some feedbacks between the original implementation and the modified(flipped) using the Vote & Flip algorithm.

Table 5. Modified polarity words in sentiment dictionary

Word	Original polarity	Modified polarity
Fun	Both positive and negative	Positive
Fine	Both positive and negative	Positive
Miss	Negative	Neutral
Challenge	Negative	Positive
Extreme	Negative	Neutral
Thumb	Negative	Neutral
Object	Negative	Neutral
Overcome	Negative	Positive
Lecture	Negative	Neutral
Negative	Negative	Neutral
Concern	Negative	Positive

4.4. Adaptation using Vote & Flip algorithm

After calculate the sentiment score, we use the *Vote & Flip* algorithm to detect negating words and change the polarity of feedback if needed. If feedback don't have any negating words, so will returned the original polarity previous classified by the sentiment score. Table 6 shows a comparison between the results obtained with the original implementation and the modified implementation(flipped) with Vote & Flip algorithm, using some feedbacks published in the original article.

Almost all feedbacks achieved the same score, except the second, where sentiment score is 1 and flipped is 0(both are wrong, correct polarity is positive), and the last feedback, where sentiment score is 1 and flipped score is -1(this time flipped score is correct, since the correct polarity is negative).

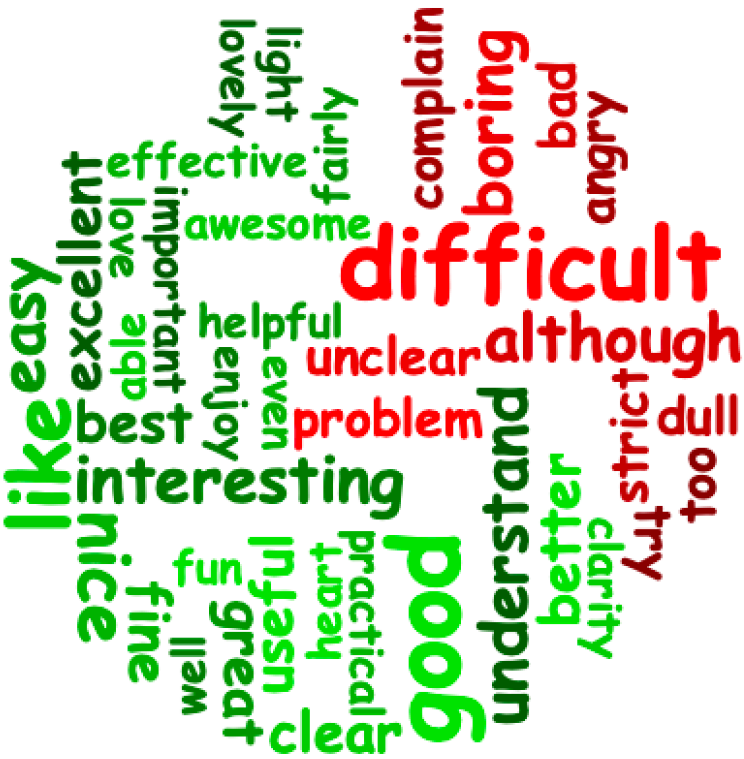

Figure 3. Word cloud created from student's feedback

Table 6. Samples of student's textual feedback

Student's feedback	Sentiment Score	Flipped Score	Actual Sentiment
Good instructor, a very nice person	2	2	Positive
The topic is always well explained and the teacher never gets angry for asking too many questions	-1	0	Positive
A good teacher	1	1	Positive
Lab was difficult because of size of students. Nobody knows what is going in front portion of class	-1	-1	Negative
Explains easy questions in class and is giving extreme difficult questions in exam	0	0	Negative
The number of students in class is high due to which teacher faces problem in paying attention to every student	-1	-1	Negative
Good effort by teacher. Some concepts are unclear, needs to be explained in more detail	0	0	Mixed
This course is very important for all CS students but the teacher needs to change her casual behaviour(most of our classes started late)	1	-1	Negative

Tables 7 and 8 shows a confusion matrix with both implementations, with a sample of 51 feedbacks, using a mix of all feedbacks published in the original article and some feedbacks collected from other sources that are similar to the published ones. Table 9 shows an accuracy's comparison of implementations. There we can see a small improvement in classification accuracy using the Vote & Flip algorithm, where the original implementation classified 76% of the feedbacks correctly, against 80% with Vote & Flip.

Table 7. Classification of feedbacks using the original implementation

		Predicted			
		Positive	Negative	Neutral	Total
Actual	Positive	33	2	2	37
	Negative	4	3	2	9
	Neutral		2	3	5
	Total	37	7	7	51

Table 8. Classification of feedbacks using vote & flip algorithm

		Predicted			
		Positive	Negative	Neutral	Total
Actual	Positive	30	3	4	37
	Negative	1	6	2	9
	Neutral			5	5
	Total	31	9	11	51

Table 9. Accuracy comparison from original implementation and vote & flip

	Accuracy
Original implementation	0,76
Vote & Flip	0,80

5. Conclusion

In this article we re-implemented the solution proposed by [Rajput, Haider and Ghani 2016] for sentiment analysis of student's feedback for the teacher's evaluation process. Additionally, we made an adaptation in the original solution to use the Vote & Flip algorithm, to detect context changes from negating words. The proposed change demonstrated a small accuracy improvement over the original solution. The source code of this experiment, can be found at: https://github.com/Oliveirakun/lexicon-sentiment-analysis.

As future work, we could combine this technique with machine learning algorithms like SVM or Naive Bayes, that demonstrated some good results in experiments made by other authors. Also, we could explore an approach using n-grams and adding more words to the dictionary, that maybe, could improve even more the accuracy of the algorithm.

References

Rajput, Q., Haider, S. and Ghani, S. (2016) "Lexicon-Based sentiment analysis of teacher's evaluation", In: Applied Computational Intelligence and Soft Computing, Hindawi Publishing Corporation.

MacKim, S. and Calvo, R. A. (2010) "Sentiment analysis in student experiences of learning," In: Proceedings of the 3rd International Conference on Educational Data Mining (EDM '10), pp. 111–120, Pittsburgh, Pa, USA.

Jagtap, B. and Dhotre, V. (2014) "SVM & HMM based hybrid approach of sentiment analysis for teacher feedback assessment," International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 3, no. 3, pp. 229–232.

Ogden, J. and Lo, J. (2010) "How meaningful are data from Kim, S. M. & Calvo, R. A. Sentiment analysis in student experiences of learning," in Proceedings of the 3rd International Conference on Educational Data Mining, Pittsburgh, Pa, USA.

Medhat, W., Hassan, A. and Korashy, H. (2014) "Sentiment analysis algorithms and applications: a survey," Ain Shams Engineering Journal, vol. 5, pp. 1093–1113.

Choi, Y. and Cardie, C. (2009) "Adapting a Polarity Lexicon using integer linear programming for domain-specific sentiment classification" EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2 Pages 590–598.

Kim, S. and Hovy, E. (2004) "Determining the sentiment of opinions," in Proceedings of the 20th International Conference on Computational Linguistics (COLING '04), Association for Computational Linguistics.

Leong, K., Lee, H. and Mak, W. (2012) "Mining sentiments in SMS texts for teaching evaluation," Expert Systems with Applications, vol. 39, no. 3, pp. 2584–2589

Goeuriot, L., Kelly, L., Jones, F., Muller, H. and Zobel, J. (2014) "Report on the SIGIR 2014 Workshop on Medical Information Retrieval (MedIR)," ACM SIGIR Forum, vol. 48, no. 2, pp. 78–82.

Altrabsheh, N., Gaber, M. and Cocea, M. (2013) "SA-E: sentiment analysis for education," in Proceedings of the 5th KES International Conference on Intelligent Decision Technologies (KES-IDT '13), Sesimbra, Portugal.

Wilson, T., Wiebe, J. And Hoffman, P. (2005) "Recognizing contextual polarity in phrase-level sentiment analysis" in Proceedings of the Human Language Technology(HLT/EMNLP '05), pp. 347–354, Stroudsburg, Pa, USA.

Cambria, E. (2013) "An introduction to concept-level sentiment analysis", in Advances in Soft Computing and Its Applications, Lecture Notes in Computer Science, pp. 478-483, Springer, Berlim, Germany