Singapore Management University
Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

9-2017

Inferring spread of readers' emotion affected by online news

Agus SULISTYA
Singapore Management University, aguss.2014@phdis.smu.edu.sg

Ferdian THUNG
Singapore Management University, ferdiant.2013@phdis.smu.edu.sg

David LO
Singapore Management University, davidlo@smu.edu.sg

DOI: https://doi.org/10.1007/978-3-319-67217-5_26

Citation

SULISTYA, Agus; THUNG, Ferdian; and LO, David. Inferring spread of readers' emotion affected by online news. (2017). *Social informatics: 9th International Conference, SocInfo 2017, Oxford, UK, September 13-15: Proceedings.* 10540, 426-439. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/3961

Published in Social Informatics: SocInfo 2017. Lecture Notes in Computer Science, vol 10540, pp. 426-439
https://doi.org/10.1007/978-3-319-67217-5_26

Inferring Spread of Readers' Emotion Affected by Online News

Agus Sulistya[1,2](✉), Ferdian Thung[1], and David Lo[1]

[1] School of Information Systems, Singapore Management University,
Singapore, Singapore
{aguss.2014,ferdiant.2013,davidlo}@smu.edu.sg
[2] Human Capital Center, PT Telekomunikasi Indonesia, Bandung, Indonesia

Abstract. Depending on the reader, A news article may be viewed from many different perspectives, thus triggering different (and possibly contradicting) emotions. In this paper, we formulate a problem of predicting readers' emotion distribution affected by a news article. Our approach analyzes affective annotations provided by readers of news articles taken from a non-English online news site. We create a new corpus from the annotated articles, and build a domain-specific emotion lexicon and word embedding features. We finally construct a multi-target regression model from a set of features extracted from online news articles. Our experiments show that by combining lexicon and word embedding features, our regression model is able to predict the emotion distribution with RMSE scores between 0.067 to 0.232 for each emotion category.

Keywords: Social emotion · Multi target regression · Machine learning

1 Introduction

Nowadays, online news platforms are popular due to their up-to-date contents, and have become important sources of information. The platforms provide a convenient way for publishers to share latest news that can quickly reach online readers. The platforms also allow readers to interact by providing comments and votes, and by sharing news articles on social media.

Publishers, writers, and many others can potentially benefit from news readers' responses. The responses can be used to measure degree of user engagement. More comments or feedbacks given by readers indicate higher popularity of a news article. The responses can also be used as a clue for placing advertisement. Moreover, readers' responses can help publishers, writers, individuals, and organizations to learn how a certain issue is viewed by the public in general. Such insight can potentially be used by decision makers to make more informed decisions (e.g., on policy and business strategy). Given the value of readers' responses, it would be beneficial to be able to predict *early* how public are likely to respond to a particular issue described in a news article.

A news article should be objective as it is intended only to report facts. This means that readers' opinions to a news article is not contained in the article itself.

To give an impression of objectivity, the writers often avoid using overly positive or negative vocabulary, or resort to other means to express their opinion, such as embedding statements in a more complex discourse or argument structure, and quoting other persons who said what they feel [2]. Separate responses to the news, when available, contain readers' opinions and emotions toward the content of the news.

Predicting readers' emotion for a particular article is an emerging research area. Most studies on predicting readers' emotion translate the task into a classification problem, either by considering it as a multi-class classification (assign an article into one of the emotion categories) [7,9,11,19] or a multi-label classification (assign to each articles a set of emotion categories) [21] problem. In this paper, we formulate the problem as a multi-target regression with the goal of predicting readers' *emotion distribution*. By knowing the predicted emotion distribution, we can get a deeper insight on likely readers' responses, e.g., estimated proportion of readers who are happy with a piece of news.

We explore lexicon-based and word-vector-based features, and input them to a regression model to predict emotion distribution. As a case study, we use a popular Indonesian online news, namely detik.com. Our work complements existing works on readers' emotion analysis and prediction. Specifically, our contributions are as follows:

1. We create a new corpus consisting of Indonesian news articles for predicting readers' emotion distribution affected by news articles.
2. We compare the effectiveness of using different parts of news articles (headlines only, contents only, and both headlines and contents) to predict the spread of readers' emotion.
3. We compare the effectiveness of *domain-specific* emotion lexicon and word embeddings with *general purpose* lexicon and word embeddings for the problem of predicting emotion distribution of a news article readers.

The structure of the remainder of this paper is as follows. Related work is presented in Sect. 2. In Sect. 3, we describe the methodology of our proposed approach which consists of 5 main steps: corpus creation, word vector construction, emotion lexicon formation, feature extraction, and regression model learning. We describe our experiments and evaluate the effectiveness of our proposed approach in Sect. 4. Threats to validity is discussed in Sect. 5. We finally conclude and mention future work in Sect. 6.

2 Related Work

Predicting social emotions has been studied in past few years. Most existing works focus on building emotion lexicons or devising prediction algorithm. In this section, we briefly summarize research efforts conducted on these two focuses.

Building Emotion Lexicon. A popular resource for emotion lexicon is WordNetAffect [20], which contains manually assigned affective labels (anger, joy, etc.) to WordNet synsets (i.e., set of synonyms). AffectNet, which is a part of SenticNet project [5], contains around 10,000 words taken from ConceptNet and aligned with WordNetAffect. AffectNet maps common sense knowledge to affective knowledge (i.e., WordNetAffect affective labels). Another popular resource is NRC-EmoLex [13], which consists of 10,000 lemmas (i.e., a base word form that is indexed in the lexicon) annotated with an intensity label for each emotion. These data are manually labeled by multiple annotators. Another approach for building lexicons is through automated means. Staiano and Guerini presented DepecheMood [18], an emotion lexicon that is built by harvesting crowd-sourced affective annotation from a social news network. In the domain of non-English lexicon, Abdaoui et al. [1] built a French lexicon by performing semi-automatic translation and synonym expansion for words in NRC-EmoLex. Nguyen et al. [14] proposed an approach to mine public opinions from Vietnamese text using a domain-specific sentiment dictionary that was built incrementally. In this paper, we extend Staiano and Guerini work [18]. Specifically, we automatically build emotion lexicon using affective-annotated news articles in an under-resource language (Indonesian) from a popular online news platform.

Predicting Social Emotion. SemEval-2007 [19] is considered the first research effort in predicting readers' emotions by analyzing news article headlines. It used news headlines as its data source. Rao et al. [15] proposed an algorithm and pruning strategies to automatically build a word-level emotion dictionary, in which each word is associated with a distribution of social emotions. They also proposed to use topic modeling for constructing a topic-level emotion dictionary, in which each topic is associated with a distribution of social emotions. Lei et al. [9] proposed an approach that performs document selection, Part-Of-Speech (POS) tagging, and social emotion lexicon generation system to detect social emotion. Hsieh et al. [7] proposed a document modeling method that utilizes embeddings of emotion keywords to perform readers' emotion classification. Recent work by Bandhakavi et al. [3] compared General Purpose Emotion Lexicons (GPELs) and Domain-Specific Emotion Lexicons (DSELs) for emotion detection from text. They confirmed the superiority of DSELs for emotion detection. Lin and Chen [10] proposed the use of a regression model to estimate readers' emotion towards news article. They use Chinese character bi-gram, Chinese words, and news metadata as features, and use Support Vector Regression (SVR) as the regression model. Our work complements these research efforts by exploring different sets of features by combining word vectors and emotion lexicon generated from different parts of news articles (headlines, contents and both).

3 Methodology

Overall framework of our approach is depicted in Fig. 1. In *Construct Corpus* step, we collect a set of online news' links that are mentioned in Twitter, and crawl the corresponding news headlines and contents to build our news article corpus. By analyzing the corpus, we build emotion lexicon and train word vectors in *Build Emotion Lexicon* and *Build Word Vector* step, respectively. We

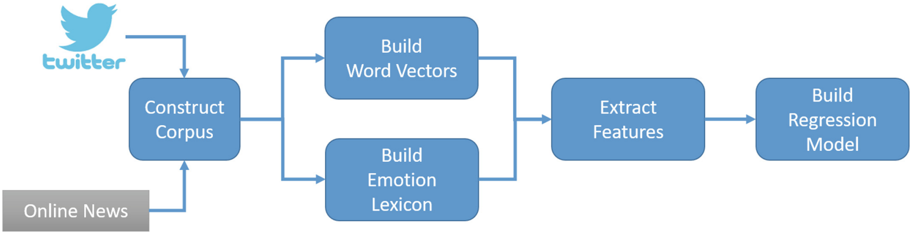

Fig. 1. Our approach's overall framework to predict readers' emotion distribution

extract features based on emotion lexicon and word vectors in *Extract Features* step. In *Build Regression Model* step, we use different combinations of extracted features to build regression models that predict reader's emotion distribution. We elaborate the above-mentioned steps in the following subsections.

3.1 Problem Definition

In this paper, we aim to predict readers' emotion distribution affected by reading a news article. Given a corpus of documents D, with their emotion scores E, where E_i is the emotion score vector for a news article D_i, we want to predict emotion scores E' for a set of new articles D' . An example of a document-emotion score vector of a particular article is: ⟨ happy:0.4; amused:0.0; inspired:0.0; dont_care:0.6; sad:0.0; afraid:0.0; angry:0.0 ⟩.

3.2 Step 1: Construct Corpus

Many news organizations have recognized the potential of social media and have used social media marketing to attract online audiences; for example, by using Twitter to promote certain articles that might interest their readers. Therefore, we are interested in an online news platform that actively tweets news article headlines, and also provide emotion scoring and commenting features for the readers.

We identify an online news platform in Indonesia, namely detik.com. According to Alexa web ranking[1], it is ranked as the most popular online news and the fourth most popular website in Indonesia. The news platform provides features that allow users to give emotion score to a particular article, as shown in Fig. 2. There are eight different emotion categories, which can be translated in English as: *Happy, Amused, Inspired, Don't Care, Annoyed, Sad, Afraid* and *Angry*. The emotion score for each category will be shown in the same page as the article that the scores correspond to.

The online news platform (detik.com) also has a Twitter account: *detikcom*. It has a huge number of followers (13.7 millions as of April 2017) and ranked

[1] http://www.alexa.com/topsites/countries/ID.

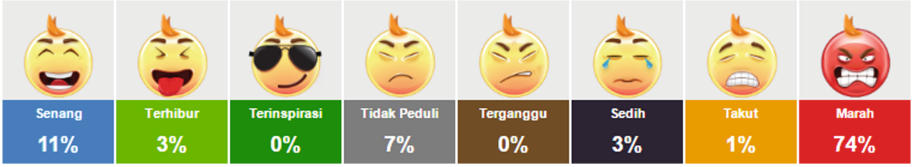

Fig. 2. A sample of emotion scores of an article published in online news

number 3 in Indonesia in terms of number of followers[2]. This makes detik.com a good source of data for analyzing the sentiment distribution of news articles' readers.

We initially collect news articles that were mentioned in detik.com's Twitter account from November 2016–February 2017. We use Python Tweepy[3] module to get its Twitter timeline. We find that there are many duplicate tweets that refer to the same news article. Online media tend to repost the same content in order to get more traffic, hit multiple time zones, and reach new followers. We remove duplicate tweets by keeping the earliest tweet and removing newer ones. After removing duplicate tweets, we have 36,587 distinct tweets.

We then process the tweets, get their contents, numbers of retweets, and favorite counts. To get the corresponding articles from the online news platform, we extract URLs from the tweets. We build a custom webpage scraper and download the articles pointed to by the URLs. For each article, we get its headline, content and emotion scores.

We remove news articles that have no emotion scores; after this step, we have 11,704 news articles. However, some of these articles may only receive very few emotion votes. We further remove articles that are likely to receive few emotion votes. We use number of comments as a proxy to number of votes[4]. We believe that the number of comments should be less than the number of votes, since it is more difficult and time consuming to write a comment, as compared to providing a vote. Therefore, we further filter our dataset to exclude articles that have less than 10 comments. At the end of this step, we have 1,575 articles remaining as our final dataset. Since we want to predict emotion distribution of unseen documents, we order the dataset based on the article's date, and use 80% from the ordered data as our training corpus. This corpus is used for building emotion lexicon and word vector features. The remaining 20% of the ordered set is used as our testing corpus. This corpus is used to evaluate the performance of emotion distribution prediction approaches.

Table 1 reports the average proportion of votes for each emotion for articles in our detik.com dataset. Note that “happy‘’ emotion has a higher score (i.e., most articles receive higher proportion of “happy” votes than other votes) than other

[2] https://www.socialbakers.com/statistics/twitter/profiles/indonesia/.

[3] http://www.tweepy.org/.

[4] The website does not provide the number of votes but only the proportion of votes for the various emotions.

Table 1. Average emotion scores from our detik.com dataset

Sentiment	Mean score
Happy	0.41
Amused	0.05
Inspired	0.05
Don't Care	0.18
Annoyed	0.04
Sad	0.09
Afraid	0.02
Angry	0.18

emotions. Possible explanations for this observation is due to characteristics of commenters, or our dataset selection process. The predominance of "happy" emotion has also been found in other datasets used in a related work by Staiano and Guerini [18].

3.3 Step 2: Build Word Vector

Word embedding is a technique to represent words in a form of continuous value vectors. These vectors encode meanings of words. One of the most popular word embedding technique is *word2vec*. *word2vec* uses a shallow neural network to reconstruct contexts of words. Two architectures can be used to generate the vectors: continuous bag-of-words (CBOW) or continuous skip-gram (SG) [12]. For CBOW, a neural network is trained to predict a word based on its surrounding words. In this architecture, the continuous value vector for a word is the vector that is input to the last layer in the network after we input its surrounding words to the network. For SG, a neural network is trained to predict surrounding words based on the current word. In this architecture, the continuous value vector for a word is the vector that is output by the first layer in the network.

Continuous value vectors that are generated by *word2vec* contain semantic meanings of words. Words that appear in similar contexts tend to have similar vector representations. The vectors also have an interesting arithmetic feature. For example, the resultant vector of the following arithmetic operation (vector of brother − vector of man + vector of woman) is pretty similar to the vector of sister. This is related to analogical reasoning where brother is to sister as man is to woman, which is encoded in the vector representation learned by *word2vec*.

Building on top of the success of word embeddings, we learn a custom word embedding from our training corpus. In practice, SG tends to be more effective than CBOW when larger datasets are available [8]. However, due to relatively small size of our training corpus, we use CBOW model to build word vectors. To create word vectors, we first split news articles in the corpus into sentences.

Indonesian texts use the same sentence end symbols as those used in English texts (sentences can end with “?”, “!”, or “.”). We use NLTK’s punkt tokenizer[5] for sentence splitting. Given the generated sentences, we train *word2vec* model using Python’s gensim module [16]. We compute 300-dimensional word embeddings with CBOW model on our training corpus, without removing stop words. We have 76,752 word vectors generated from our training set.

3.4 Step 3: Build Emotion Lexicon

Emotion lexicon is a dictionary that associates words with emotion categories, such as anger, fear, surprise, sadness, etc. It is typically constructed via crowdsourcing. In the crowdsourcing process, a group of people is asked to label a set of words by associating each word with one or more basic emotions. Labels from the group of people are then aggregated for each word by summing up vote for each basic emotion. The resultant sums are then normalized across basic emotions, which represent emotion distribution for the corresponding word. The resultant collection of words along with their corresponding emotion distribution is the constructed emotion lexicon.

Another approach to create emotion lexicon is to use a crowd-sourced affective annotation from a social news network, such as the one used in Staiano and Guerini work [18]. Typically, an automated tools such as web crawler is used to get news articles and related emotion scores tagged by readers. By splitting a news article into words, emotion scores for each word in the article are calculated.

To create an emotion lexicon, we first construct a document-by-emotion matrix containing the eight emotion scores for each document. We follow a previous work to create a word-by-emotion matrix [18]. We also create a word-by-document matrix containing normalized word frequency across documents. We multiply the document-by-emotion matrix and the word-by-document matrix to produce a word-by-emotion matrix. In the end, we have 22,346 words and their corresponding eight emotion scores that we refer to as our generated Emolex (Emotion Lexicon). An excerpt of the matrix is shown in Table 2.

Table 2. Sample taken from *word-by-emotion* matrix generated by analyzing our detik.com training corpus

Word	Happy	Amused	Inspired	Don’t Care	Annoyed	Sad	Afraid	Angry
Walikota (Mayor)	0.488	0.032	0.087	0.209	0.010	0.028	0.006	0.142
Membunuh (Kill)	0.246	0.038	0.091	0.055	0.017	0.152	0.047	0.354
Pahlawan (Hero)	0.442	0.050	0.051	0.058	0.040	0.080	0.016	0.264

[5] http://www.nltk.org/.

3.5 Step 4: Extract Features

A news article contains headline and content. We explore different combinations of news article parts to extract features from, i.e., use only headlines, contents or both. We follow emotion lexicon construction process (described in Sect. 3.4) and word vectors training process (described in Sect. 3.3).

Lexicon Features. We build around 22,000 lexicons tagged with emotion scores as described in Sect. 3.4. We transform each news article in our corpus into a document-by-emotion feature vector by following these steps:

1. We split the considered portion of a news article into words, and then remove the stop words.
2. For each word, we retrieve its emotion score vector from our word-by-emotion matrix.
3. We calculate the emotion vector for the news article by averaging emotion vectors of the words in the news article.

In the end, we have a document-by-emotion matrix of the following dimension: total number of articles in the corpus (1,575) × emotion scores (8). Each document-emotion vector in the matrix represents emotion lexicon features for the corresponding news article.

Word Vector Features. Our set of trained word vectors model include around 76,000 vectors of 300 dimensions. We generate a vector for each news article. To do this, we use a vector-averaging approach, which consists of the following steps:

1. We split the considered portion of a news article into words, and then remove the stop words.
2. For each word, we retrieve its word vector from our trained *word2vec* model.
3. We generate a vector for the news article by averaging word vectors that corresponds to the words in the news article.

As a result, we have a vector for each news article in the corpus. Since we have 1,575 news articles, we get $1{,}575 \times 300$ matrix.

3.6 Step 5: Build Regression Model

We formulate our problem as a multi-target regression task. Multi-target regression is a family of regression techniques where there are multiple output variables. In multi-target regression task, a set of training example E is given, where each example is in the form of $(\mathbf{x}, \mathbf{y})$. $\mathbf{x} = \{x_1, x_2, x_3, ..., x_A\}$ is a vector of A attributes and $\mathbf{y} = \{y_1, y_2, y_3, ..., y_T\}$ is a vector of T target values. Multi-target regression learns a model that, given $\mathbf{x}$, predicts all T target values in $\mathbf{y}$ simultaneously. Multi target regression is generally solved by transforming it to multiple single-target regression or adapting the regression algorithm to directly deal with multiple outputs.

Given features extracted from our training corpus, we build a multi-target regression model to predict spread of readers' emotions. We explore different sets of features, i.e. emotion lexicon features (with 8 independent variables), word vector features (with 300 variables), and combination of both.

4 Experiments and Results

In this section, we first describe our experiment setting, baselines used and evaluation metrics. Then, we present our research questions and results of our experiments which answer the questions.

4.1 Experiment Setting and Evaluation

Experiment Setting. Our dataset consists of 1,575 articles. We use 80% of this data as training corpus, and the remaining as testing corpus. Before we build word vector model as described in Sect. 3.3, we preprocess the corpus using Python NLTK and Scikit module. We remove punctuations and non-word characters, and convert the remaining characters into lowercase.

We use Scikit-Learn[6] module to build a multi-target regression model. The module supports multi-target regression by transforming it into multiple single-target regression tasks. The single-target regression algorithm is determined by choosing a base regressor. For choosing a good base regressor, we experimented with different regressors such as Linear Model, Random Forest Regressor, Support Vector Regressor, and Gradient Boosted Regressor. We found that Gradient Boosted regressor achieves the best overall performance compared to other regressors. Therefore, we use this regressor in our experiments.

All experiments were done on an Intel Core i7 CPU, 8 GB RAM notebook running Windows 10 64 bit.

Baseline. We compare our model with two general purpose models that can be used for emotion prediction:

1. We use Sentic-API [6] as a general purpose emotion lexicon. Sentic-API supports Indonesian language. It contains denotative (i.e., semantics) and connotative information (sentics) associated with 50,000 common sense concepts. A word-emotion lexicon that associates words with four dimensions of sentics (pleasantness, attention, sensitivity, and aptitude) is also provided. For each news article, we take the sentics values for each word and compute the average value for each sentics dimension. The generated four average values (i.e., each corresponding to a particular sentics dimension) is the news sentics. These average values are converted to a feature vector that is input to a multi-target regression model. An excerpt of the word-sentics matrix is shown in Table 3.

[6] http://scikit-learn.org.

Table 3. An excerpt of Sentic-API's word-sentics matrix

Word	Aptitude	Attention	Pleasantness	Sensitivity
Gembira (Happy)	0.193	0.156	0.504	−0.176
Sedih (Sad)	−0.051	0.266	−0.826	−0.461
Walikota (Mayor)	0.000	0.152	0.079	−0.061

2. We use a freely available word vector model trained using FastText [4] as the general purpose word vector. This model is trained on Wikipedia dataset, and it is available for 294 languages including Indonesian. The pre-trained model has word vectors with dimension of 300, and were obtained using the skip-gram model described in Bojanowski et al.'s paper [4]. For each news article, the word vector associated with each word is collected and averaged. The averaged word vector is considered as the news representation and input to a multi-target regression model.

Evaluation. To measure the effectiveness of our approach and the baselines, we use RMSE (Root Mean Squared Error). RMSE is a widely used evaluation metric when estimating continuous values. It is the square root of the average of squared differences between prediction and actual observation. The metric is defined below:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

where N is the number of documents, y_i is the ground truth of emotion score, and $\hat{y}_i$ is the predicted emotion score.

4.2 Research Questions and Results

RQ1: How effective is the use of different portions of news article (headlines only, contents only, headlines+contents) in predicting emotion scores?

Approach: In this research question, we investigate the effectiveness of using three different portions of news articles: news headlines only, news contents only and combination of news headlines and contents. For each of them, we trained separate *word2vec* models using gensim. We also create different word-by-emotion matrix. Based on the extracted features, we build regression models and calculate RMSE for each emotion category.

Results: Table 4 shows the results of our experiments. We can see that generally, using headline combined with contents performs at least as good as using headline

Table 4. RMSE scores of the emotion lexicon (EM) and word vectors (WV) when considering different portions of news articles: Headlines (H), Contents (C), and Headlines+Contents (H+C)

	Features	Happy	Amused	Inspired	Don't Care	Annoyed	Sad	Afraid	Angry	Average
H	WV	0.299	0.105	0.105	0.230	0.068	0.149	0.039	0.259	0.157
C	WV	0.284	0.107	0.113	0.235	0.063	0.148	0.058	0.243	0.156
H+C	WV	0.278	0.098	0.120	0.213	0.060	0.145	0.077	0.252	0.155
H	EM	0.308	0.071	0.118	0.242	0.071	0.143	0.051	0.239	0.155
C	EM	0.299	0.079	0.111	0.257	0.046	0.142	0.056	0.252	0.155
H+C	EM	0.277	0.103	0.095	0.203	0.056	0.131	0.034	0.216	0.151

or content only. This finding suggests that both headlines and contents contain useful information that can be combined together to create a better model.

RQ2: How effective are the generated emotion lexicon (EM) and word vectors (WV) as compared to the general purpose baselines?

Approach: In this research question, we compare the effectiveness of using emotion lexicon and word vector generated by our approach against the general purpose baselines (see Sect. 4.1) to predict emotion scores distribution. Our previous experiment shows that using news headline combined with news content generally produces better result. Therefore, we use this combination for this experiment.

Results: Table 5 shows the results of our experiments. Our generated emotion lexicon features achieve better performance for predicting scores in all emotion categories, when compared to using Sentics. Similarly, our generated word vectors achieves a better performance, as compared to using a general pre-trained word vector from Wikipedia using FastText. Comparing average RMSE over all emotions, EM outperforms Sentics by 14.7%, while WV outperforms FastText by 84.9%. These results show the usefulness of building domain specific emotion lexicon and training domain specific word vectors.

Table 5. RMSE scores of our generated emotion lexicon (EM) and word vectors (WV) as compared to a general purpose lexicon (Sentics) and word vectors trained on Wikipedia (FastText) on predicting emotion distribution scores

Features	Happy	Amused	Inspired	Don't Care	Annoyed	Sad	Afraid	Angry	Average
EM	0.277	0.103	0.095	0.203	0.056	0.131	0.034	0.216	0.151
Sentics	0.328	0.105	0.114	0.297	0.059	0.154	0.083	0.272	0.177
WV	0.278	0.098	0.120	0.213	0.060	0.145	0.077	0.252	0.155
FastText	0.373	1.695	1.314	0.494	1.385	1.127	1.293	0.54	1.028

RQ3: Can combining emotion lexicon and word embedding vectors improve the performance of the prediction model?

Approach: To answer this question, we combine emotion lexicon and word vector features (EM+WV), and compare it with using emotion lexicon features alone (EM) and word vector features alone (WV). Similar like RQ2, we extract features from both headlines and contents of news articles.

Results: Table 6 shows the results of our experiments. By combining emotion lexicon features and word embedding vector features (EM+WV), the average RMSE score is reduced from 0.151 to 0.130 (13.91%) as compared to EM, and from 0.155 to 0.130 (16.13%) as compared to WV. Therefore, by combining emotion lexicon and word vector features, we can improve performance of the regression model.

Table 6. RMSE scores of the combination of emotion lexicon and word vector features (EM+WV) compared to when each set of features is used alone (EM or WV)

Features	Happy	Amused	Inspired	Don't Care	Annoyed	Sad	Afraid	Angry	Average
EM	0.277	0.103	0.095	0.203	0.056	0.131	0.034	0.216	0.151
WV	0.278	0.098	0.120	0.213	0.060	0.145	0.077	0.252	0.155
EM+WV	0.232	0.090	0.102	0.158	0.067	0.129	0.067	0.193	0.130
Sentics+WV	0.304	0.132	0.133	0.207	0.090	0.166	0.083	0.256	0.171

5 Threats to Validity

There are a number of threats that may affect the validity of our findings. In this section, we discuss threats to internal validity, external validity, and construct validity.

Internal Validity. Threats to internal validity relates to experimenter bias and errors in our implementation. We have checked our implementation, but there could still be errors that we do not notice.

External Validity. Threats to external validity relate to the generalizability of our findings. We have evaluated the effectiveness of our approach to infer readers' emotion scores in a corpus of 1,575 online news articles. In the future, we plan to reduce this threat further by considering a larger set of articles from various online news platforms.

Construct Validity. Threats to construct validity relate to the suitability of our evaluation metric. In this work, we use RMSE as the evaluation metric. RMSE is a standard metric and it has been used as evaluation metrics in past studies such as in [10]. Thus, we believe that threats to construct validity are minimal.

6 Conclusion and Future Work

In this paper, we have presented an approach that use emotion lexicon and word embedding in order to predict readers' emotion scores distribution towards an online news article. We build a new corpus containing around 1,5k Indonesian news articles taken from detik.com along with affective annotations provided by readers of those articles. Our experiments show that, by using combined features of domain-specific emotion lexicon together with word embeddings vectors, we are able to predict the distribution of readers' emotion scores with a Root Mean Squared Error (RMSE) score ranging from 0.067 to 0.232. Our approach is generic and can be easily replicated to other online news platforms that allow readers to provide affective annotations. Our approach can benefit publishers by giving them early insight on expected public response to a particular article, before they actually publish it.

In the future, we plan to evaluate our proposed approach on another corpus. To improve the accuracy of our approach further, we plan to experiment with more features to better characterize different reader emotions. One possibility is by improving accuracy of the emotion lexicon using bag-of-concepts [17] instead of bag-of-words.

Acknowledgments. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative, and PT Telekomunikasi Indonesia (Telkom).

References

1. Abdaoui, A., Azé, J., Bringay, S., Grabar, N., Poncelet, P.: Expertise in French health forums. Health Inf. J. 1460458216682356 (2016)
2. Balahur, A., Steinberger, R., Kabadjov, M., Zavarella, V., Van Der Goot, E., Halkia, M., Pouliquen, B., Belyaeva, J.: Sentiment analysis in the news. arXiv preprint (2013). arXiv:1309.6202
3. Bandhakavi, A., Wiratunga, N., Massie, S., Padmanabhan, D.: Lexicon generation for emotion detection from text. IEEE Intell. Syst. **32**(1), 102–108 (2017)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint (2016). arXiv:1607.04606
5. Cambria, E., Hussain, A.: Sentic Computing: Techniques, Tools, and Applications, vol. 2. Springer Science & Business Media, Netherlands (2012)
6. Cambria, E., Poria, S., Bajpai, R., Schuller, B.W.: Senticnet 4: a semantic resource for sentiment analysis based on conceptual primitives. In: COLING, pp. 2666–2677 (2016)
7. Hsieh, Y.L., Chang, Y.C., Chu, C.H., Hsu, W.L.: How do i look? Publicity mining from distributed keyword representation of socially infused news articles. In: Conference on Empirical Methods in Natural Language Processing, p. 74 (2016)
8. Lai, S., Liu, K., He, S., Zhao, J.: How to generate a good word embedding. IEEE Intell. Syst. **31**(6), 5–14 (2016)
9. Lei, J., Rao, Y., Li, Q., Quan, X., Wenyin, L.: Towards building a social emotion detection system for online news. Future Gener. Comput. Syst. **37**, 438–448 (2014)

10. Lin, K.H.Y., Chen, H.H.: Ranking reader emotions using pairwise loss minimization and emotional distribution regression. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 136–144. Association for Computational Linguistics (2008)
11. Lin, K.H.Y., Yang, C., Chen, H.H.: Emotion classification of online news articles from the reader's perspective. In: 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2008, vol. 1, pp. 220–226. IEEE (2008)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint (2013). arXiv:1301.3781
13. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon. Comput. Intell. **29**(3), 436–465 (2013)
14. Nam Nguyen, H., Van Le, T., Son Le, H., Vu Pham, T.: Domain specific sentiment dictionary for opinion mining of vietnamese text. In: Murty, M.N., He, X., Chillarige, R.R., Weng, P. (eds.) MIWAI 2014. LNCS, vol. 8875, pp. 136–148. Springer, Cham (2014). doi:10.1007/978-3-319-13365-2_13
15. Rao, Y., Lei, J., Wenyin, L., Li, Q., Chen, M.: Building emotional dictionary for sentiment analysis of online news. World Wide Web **17**(4), 723–742 (2014)
16. Rehurek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Citeseer (2010)
17. Sahlgren, M., Cöster, R.: Using bag-of-concepts to improve the performance of support vector machines in text categorization. In: Proceedings of the 20th International Conference on Computational Linguistics, p. 487. Association for Computational Linguistics (2004)
18. Staiano, J., Guerini, M.: Depechemood: a lexicon for emotion analysis from crowd-annotated news. arXiv preprint (2014). arXiv:1405.1605
19. Strapparava, C., Mihalcea, R.: Semeval-2007 task 14: affective text. In: Proceedings of the 4th International Workshop on Semantic Evaluations, pp. 70–74. Association for Computational Linguistics (2007)
20. Strapparava, C., Valitutti, A., et al.: Wordnet affect: an affective extension of wordnet. In: LREC, vol. 4, pp. 1083–1086 (2004)
21. Zhang, Y., Su, L., Yang, Z., Zhao, X., Yuan, X.: Multi-label emotion tagging for online news by supervised topic model. In: Cheng, R., Cui, B., Zhang, Z., Cai, R., Xu, J. (eds.) APWeb 2015. LNCS, vol. 9313, pp. 67–79. Springer, Cham (2015). doi:10.1007/978-3-319-25255-1_6