applied sciences

MDPI

Article

Similar Word Replacement Method for Improving News Commenter Analysis

Deun Lee and Sunoh Choi *

Department of Software Engineering, Jeonbuk National University, Jeonju 54896, Korea; deunlee123@gmail.com
* Correspondence: suno7@jbnu.ac.kr

Abstract: In Korea, it is common to read and comment on news stories on portal sites. To influence public opinion, some people write comments repeatedly, some of which are similar to those posted by others. This has become a serious social issue. In our previous research, we collected approximately 2.68 million news comments posted in April 2017. We classified the political stance of each author using a deep learning model (seq2seq), and evaluated how many similar comments each user wrote, as well as how similar each comment was to those posted by other people, using the Jaccard similarity coefficient. However, as our previous model used Jaccard's similarity only, the meaning of the comments was not considered. To solve this problem, we propose similar word replacement (SWR) using word2vec and a method to analyze the similarity between user comments and classify the political stance of each user. In this study, we showed that when our model used SWR rather than Jaccard's similarity, its ability to detect similarity between comments increased 3.2 times, and the accuracy of political stance classification improved by 6%.

Keywords: internet news; user analysis; word2vec

Citation: Lee, D.; Choi, S. Similar Word Replacement Method for Improving News Commenter Analysis. *Appl. Sci.* **2022**, *12*, 6803. https://doi.org/10.3390/app12136803

Academic Editor: Rafael Valencia-Garcia

Received: 26 May 2022
Accepted: 2 July 2022
Published: 5 July 2022

1. Introduction

A survey by Korea's Ministry of Culture, Sports and Tourism found that 90% of Koreans read news on portal sites such as Naver and Daum [1–3], and write comments about these stories. Some people try to affect public opinion through their comments. In 2012, staff members at Korea's National Information Service tried to impact public opinion by writing comments in response to news articles [4]. In 2017, a person posting under the name Druking wrote comments automatically using the Kingcrap system [5]. In 2022, a political party developed a system (Kraken) to detect malicious comments [6]. As comments on news articles can affect public opinion, there is a need for a system capable of collecting such comments and analyzing their authors.

In our previous research [7], we collected approximately 2.68 million comments on news stories from April 2017. These comments were written by about 200,000 individuals regarding around 27,000 news stories, largely related to the Korean presidential election which took place in May 2017. In that study, we proposed three methods of analyzing online news commenters. First, we classified each news commenter's political stance using a seq2seq model [8]. Second, we evaluated how many similar comments each user wrote using Jaccard's similarity coefficient (JSC) [9]. Third, we evaluated how similar each user's comments were to those written by other users. Our three questions are summarized as follows:

$$\begin{aligned} Q_1 &: \text{What is the political stance of the news commenter?} \\ Q_2 &: \text{How many similar comments does each news commenter write?} \\ Q_3 &: \text{How similar are each news commenter's comments compared with those of others?} \end{aligned} \quad (1)$$

However, in our previous model using JSC, the meaning of each word in the comments was not considered. There were three comments as follows:

$$\begin{aligned} C_1 &: \text{I love you} \\ C_2 &: \text{I like you} \\ C_3 &: \text{I hate you} \end{aligned} \tag{2}$$

C_1 and C_2 have similar meanings. However, C_1 and C_3 have different meanings. When we used JSC for similarity comparison, the similarity score between C_1 and C_2 was the same as the similarity score between C_1 and C_3. This was a problem in our previous study.

Therefore, a new question is presented as follows:

Q_4: How can we consider the meaning when comparing two news comments? (3)

To achieve this goal, we propose using similar word replacement (SWR), via word2vec [10], to consider each word's similarity and classify each user's political stance. Figure 1 shows that words used by individuals whose political stance is left-leaning are similar, and words used by individuals whose political stance is right-leaning are similar. In Figure 1, the two blue points indicate 이명박 (Lee Myung-bak) and 타살 (murder), which are used by the 'political left.' These words imply that former president (Roh Moo-hyun) may have been murdered by the subsequent president (Lee Myung-bak) [11]. The two red points indicate 문빠들 (Moon supporters) and 조작댓글 (fake comments) which are used by the 'political right.' These words imply that President Moon's supporters write fake comments [5].

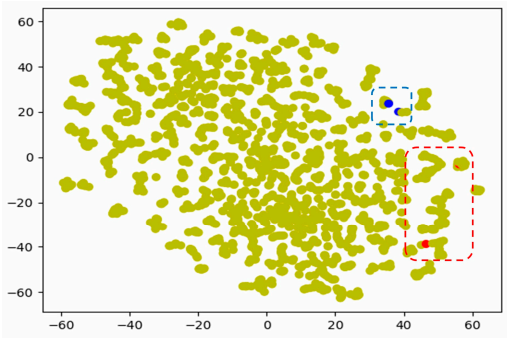

Figure 1. Word Distribution in word2vec.

Our model can determine how similar two comments are more effectively using word2vec than the Jaccard similarity coefficient. When the similarity score between two comments is computed, if a word w_i in a comment is similar to a word w_j from another comment, the word w_i is replaced with the word w_j. This process is referred to as similar word replacement (SWR).

SWR was also used to classify each user's political stance. When a word in a comment was not trained in the seq2seq model, classification of the author's political stance was difficult. When there was a new word in a comment, it was replaced with a similar word trained in the seq2seq model. This allowed the model to accurately classify each user's political stance.

In this study, first, we show that the similarity score between comments increases about 3.2 times when SWR was used compared to JSC. Second, we demonstrate that the classification accuracy of each user's political stance increased by about 6% when using SWR compared to JSC.

The remainder of this paper is structured as follows: In Section 2, we introduce related studies. In Section 3, we propose similar word replacement (SWR) as a method of classifying each commenter's political stance. In Section 4, we evaluate the performance of SWR. Finally, in Sections 5 and 6 we provide our discussion and conclusions.

2. Related Work

A number of studies have used seq2seq to classify the meaning of sentences [12]. They classify the articles' meaning whether they have positive meaning or negative. Hamborg et al. introduced NewsMTSC, a high-quality dataset for target-dependent classification (TSC) of news articles [12].

Wikipedia [13] provides information about various topics, though this information may be biased and must therefore be removed. Recasens et al. discussed framing bias and epistemological bias, and identified their common linguistic cues [14]. Hube et al. proposed a supervised classification approach that uses an automatically created lexicon of words that imply bias [15].

Fan et al. investigated the effect of presenting factual content to influence readers' opinions, known as information bias [16]. Cho et al. proposed a method of classifying the political bias of news articles using subword tokenization [17].

Garrett suggested that the desire for opinion reinforcement may play an essential role in determining the exposure of each individual to online political information [18]. The results demonstrated that opinion-reinforcing information promotes exposure to news stories, whereas opinion-challenging information makes exposure less likely. The objective of our study differed from that of Garrett's, as we analyzed news portal users based on their comments.

In our previous study, we classified users' political stances using seq2seq and proposed methods, including Jaccard's similarity coefficient, to evaluate how many comparable comments each individual writes, and how similar the remarks are to those posted by other users [7]. However, Jaccard's similarity does not consider the meaning of words. Therefore, in this study, our seq2seq model uses similar word replacement based on word2vec to consider the meaning of comments on news stories to determine the similarity between them.

Social impact theory, proposed by Bibb Latane [19,20], consists of four basic rules that consider how individuals can be sources or targets of social influence. Social impact is the result of social forces, including the strength of the source of the impact, the immediacy of the event, and the number of sources exerting the impact. Our research is related to the social impact theory because many similar news comments have an impact on public opinion.

Bourdieu proposed social inertia [21,22]. Each person occupies a position in a social space that consists of his or her social class, social relationships, and social networks. Through the individual's engagement in the social space, he or she develops a set of behaviors, lifestyles, and habits that often serve to maintain the status quo. People are encouraged to accept the social word as it is rather than rebel against it. Our research is related to social inertia because many news users write comments similar to those of other users with the same political stance.

GloVe is a distributed word representation model developed by Stanford [23,24]. This is achieved by mapping words into a meaningful space, in which the distance between words is related to semantic similarity. Training is performed on aggregated global word–word co-occurrence statistics from a corpus, and the resulting representations show linear substructures of the word vector space.

fastText is a library for learning word embeddings and text classification created by Facebook's AI Research Lab [25,26]. The model allows the creation of unsupervised learning or supervised learning algorithms for obtaining vector representations of words. GloVe and fastText are also models for distributed word representations, similar to word2vec developed by Google. They have advantages and disadvantages compared with the word2vec used in this study.

Sitaula et al. proposed the use of three different feature extraction methods (fastText-based, domain-specific, and domain-agnostic) for the representation of tweets [27]. Shahi et al. proposed an analysis of people's sentiments using TF–IDF and fastText [28]. These studies are related to our research. However, they are different from ours because our

goal is to classify news commenters' political stances and evaluate how similar their comments are.

3. Similar Word Replacement for Internet News User Analysis

3.1. Necessity of Similar Word Replacement

In previous research [7], our model used Jaccard's similarity coefficient to evaluate how many similar comments each user wrote, and how comparable each remark was to those written by other users. To compare two comments, C_1 and C_2, the Jaccard similarity coefficient was computed as follows:

$$SimScore_{Origin}(C_1, C_2) = (S_1 \cap S_2)/(S_1 \cup S_2) \tag{4}$$

where S_i is a set of words used in a comment C_i.

However, Jaccard similarity has the following problem: suppose there are three comments C_1, C_2, and C_3, as follows:

$$\begin{array}{l} C_1: \text{ I love you} \\ C_2: \text{ I like you} \\ C_3: \text{ I hate you} \end{array} \tag{5}$$

The first comment has the opposite meaning of the third comment, and the first comment has a similar meaning to the second comment. However, the Jaccard similarity between C_1 and C_2 is 2/4 = 0.5, and the Jaccard similarity between C_1 and C_3 is also 2/4 = 0.5. As the Jaccard similarity coefficient does not consider the meaning of the words, even though the first comment has a similar meaning to the second comment, their Jaccard similarity is still 0.5.

To solve this problem, we propose similar word replacement (SWR) using word2vec [10]. SWR can be described as follows. The word 'like' in the second comment has a similar meaning to the word 'love' in the first comment. When computing the similarity of C_1 and C_2, if a word in C_2 is similar to a word in C_1, the word in C_2 is replaced with the similar word from C_1. C_2 is thus converted to $C_2{}'$:

$$C_2{}': \text{ I love you} \tag{6}$$

Thereafter, the similarity of C_1 and $C_2{}'$. could be computed as follows:

$$SimScore_{SWR}(C_1, C_2) = \left(S_1 \cap S_2'\right)/\left(S_1 \cup S_2'\right) = \frac{3}{3} = 1 \tag{7}$$

Therefore, when we use SWR, we know that C_2 is more similar to C_1 than C_3.

3.2. word2vec for Similar Word Replacement

word2vec [10] was used to evaluate how similar two words are to each other. word2vec predicts the current word based on the context. For example, consider the following sentence:

$$\text{I love her and you like him} \tag{8}$$

The sentence has a total of seven words as follows:

$$\{\text{I, love, her, and, you, like, him}\} \tag{9}$$

The word 'I' comes before the word 'love,' and the word 'her' comes after the word 'love.' In word2vec, the word 'I' and the word 'her' are input data, defined as follows:

$$1\,0\,1\,0\,0\,0\,0 \tag{10}$$

The word 'love' is the output data and can be defined as follows:

$$0\,1\,0\,0\,0\,0\,0 \tag{11}$$

In Figure 2, the number of nodes in the hidden layer is 5, W_{in} is a 7×5 matrix, and W_{out} is a 5×7 matrix. word2vec was able to predict the word 'love' based on the context 'I and her.'

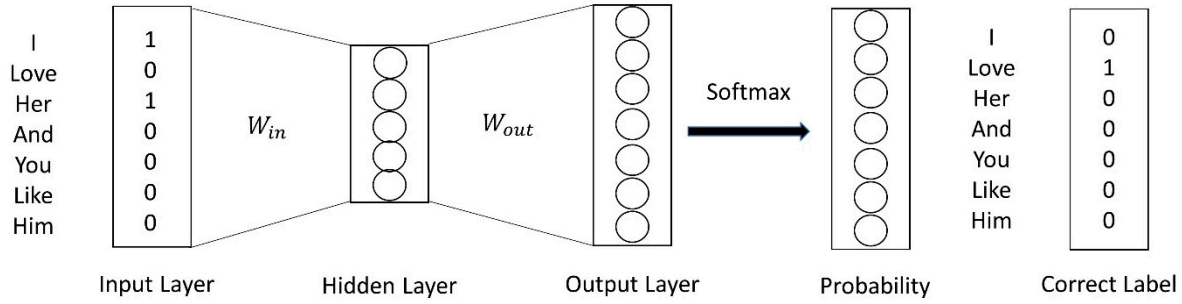

Figure 2. word2vec.

After we trained the word2vec model using the set of sentences, we get W_{in} as follows:

$$W_{in} = \begin{bmatrix} -0.90 & -1.03 & -1.46 & -1.32 & 0.93 \\ 1.21 & 1.26 & -0.07 & 0.07 & -1.23 \\ -1.08 & -0.88 & -0.33 & -0.57 & 1.04 \\ 1.02 & 1.01 & -1.62 & -1.64 & -1.05 \\ -1.06 & -0.91 & -0.31 & -0.57 & 1.04 \\ -0.90 & -1.03 & -1.46 & -1.30 & 0.92 \\ 1.09 & 1.16 & 1.43 & 1.39 & -1.07 \end{bmatrix} \tag{12}$$

Each row of W_{in} is a vector of each word in the set of sentences. For example, the vector of the word 'love' is as follows:

$$v_{love} = \{1.21,\ 1.26, -0.07,\ 0.07, -1.23\} \tag{13}$$

Our model used cosine similarity [29] to compute the similarity of two words (w_i and w_j) as follows:

$$Sim(w_i, w_j) = \frac{v_i \cdot v_j}{\|v_i\| \| v_j\|} \tag{14}$$

where v_i is a vector of a word w_i.

3.3. Similar Word Replacement Using word2vec

We then applied similar word replacement (SWR) using word2vec. Our goal was to replace a word in C_2 with a similar word in C_1. When the word in C_2 was replaced with the word from C_1, we observed increased similarity between C_1 and C_2. Thus, we can use SWR to determine if C_2 is similar to C_1.

The algorithm of SWR is given in Algorithm 1: When C_1 and C_2 are provided, we receive sets S_1 and S_2 of the words in C_1 and C_2. We then get the difference D_1 of S_1 and S_2 and the difference D_2 of S_2 and S_1.

Algorithm 1. Similar Word Replacement using word2vec

Result: The replaced comment C_2'
Let C_1 and C_2 be two comments.
$S_1 \leftarrow$ The word sets in C_1
$S_2 \leftarrow$ The word sets in C_2
$S_2' \leftarrow$ The word sets in C_2
$D_1 \leftarrow$ The difference between S_1 and S_2
$D_2 \leftarrow$ The difference between S_2 and S_1
for w_j in D_2 :
  $SW_j \leftarrow$ The similar word set of w_j
  $v_j \leftarrow$ The vector of w_j
  for w_k in SW_j :
    $v_k \leftarrow$ The vector of w_k
    $Sim(w_j, w_k) \leftarrow v_j \cdot v_k / |v_j||v_k|$
  end
  Let w_i be the word having the highest similarity
  if w_i in D_1 :
$S_2' = S_2' - w_j + w_i$
  end
end
return S_2'

When a word w_j in D_2 is given, we find a set SW_j of similar words of w_j from the word2vec database. Our model computed the similarity between w_j and a word w_k from the set SW_j of similar words.

$$Sim(w_j, w_k) \leftarrow v_j \cdot v_k / |v_j||v_k| \tag{15}$$

The model then identified a similar word w_i with the highest similarity. If the word w_i was in D_1, it replaced w_j with w_i and S_2 was converted into S_2'. Thereafter, the similarity using SWR was computed as follows:

$$SimScore_{SWR}(C_1, C_2) = \left(S_1 \cap S_2'\right) / \left(S_1 \cup S_2'\right) \tag{16}$$

As the intersection of S_1 and S_2' is equal to the union of w_i and the intersection of S_1 and S_2, the following is always satisfied:

$$SimScore_{SWR}(C_1, C_2) \geq SimScore_{Origin}(C_1, C_2) \tag{17}$$

3.4. Political Stance Classification Using SWR

SWR was used to classify the political stance of those who commented on news stories. We used SWR for the following reasons: Suppose that TWS stands for trained word set and NWS stands for new comment's word set in the seq2seq model. TW_r is a word in the TWS and NW_s is a word in the NWS.

As shown in Figure 3, our model used the word2vec database to find similar words to NW_S. If the similar word to NW_S was TW_r, NW_s was replaced with TW_r. However, if the similar word to NW_S was NW_t, it was not replaced. As NW_t was not in the TWS, it did not assist the seq2seq model. On the other hand, when NW_s was replaced with TW_r, it assisted the seq2seq model to classify political stance. In our experiment, the accuracy of political stance classification increased by about 6% when using SWR.

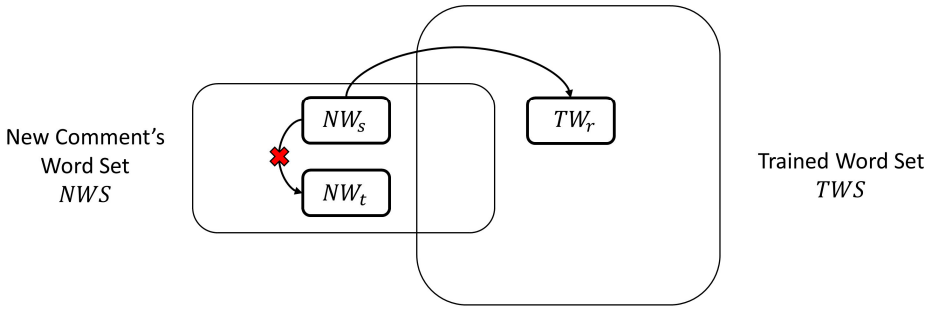

Figure 3. Similar Word Replacement for Political Stance Classification.

SWR for political stance classification is as shown in Figure 4. We generated a similar word database using word2vec. When a new comment is given, it is converted into a replaced comment using SWR. Political stance was then classified using the seq2seq model [20].

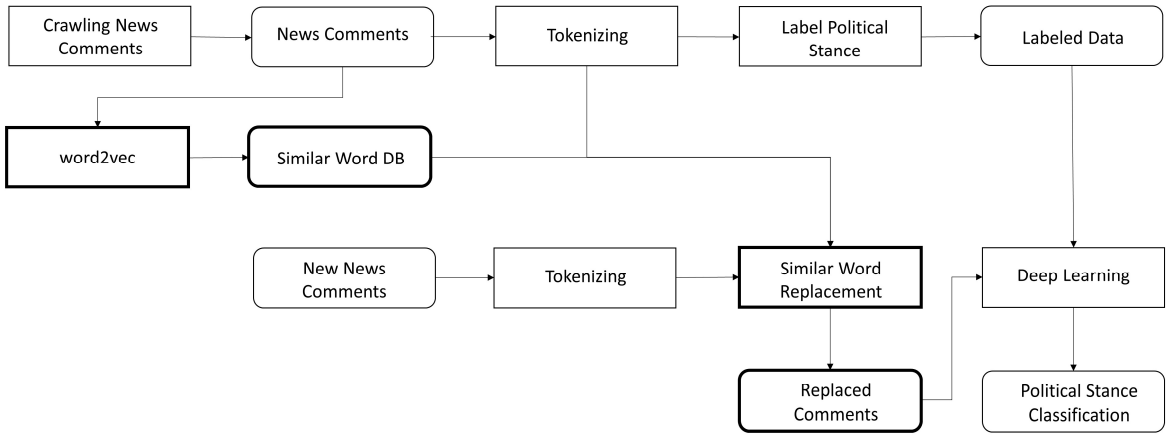

Figure 4. Political Stance Classification using SWR.

Figure 5 shows the seq2seq deep learning model used to classify each user's political stance. This model consists of an embedding layer and a long short-term memory (LSTM) layer [30]. The model's parameters are listed in Table 1.

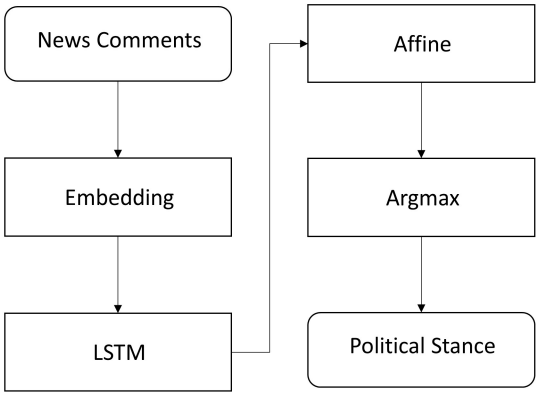

Figure 5. Seq2seq model to classify each user's political stance.

Table 1. Parameters of the proposed deep learning model.

Parameter	Value
vocab_size	2143
wordvec_size	8
hidden_size	16
batch_size	10
max_epoch	100

4. Results

In order to evaluate SWR, we performed two experiments. In Section 4.1, we introduce the experimental setup. In Section 4.2, we evaluate to what degree comment similarity increases when SWR is used. In Section 4.3, when we use SWR in order to classify each user's political stance, we evaluate how much the accuracy increases.

4.1. Experiment Environment

For the experiments, we used a computer with a 3.7 GHz i7 CPU, 16 GB of memory, an Nvidia 1080 GPU, and Windows 10 Pro operating system. In order to extract Korean words from comments, we used a Korean natural language processing (NLP) library (Hannanum) [31]. For deep learning, we used keras-gpu 2.0.8 [32].

We collected about 2.68 million comments written by approximately 200,000 users on around 27,000 news articles from April 2017, located on the website Daum [3]. In order to parse the news web pages, we used BeautifulSoup [33]. We analyzed the top 500 users of the 27,000 articles and classified their political stances for supervised learning.

We used the following accuracy metric to evaluate how effectively the deep learning model, which applied SWR, classified each user's political stance. Accuracy, recall, and false positive rate (FPR) are defined as follows:

$$\begin{aligned} \text{Accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{FN}+\text{TN}} \\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \\ \text{FPR} &= \frac{\text{FP}}{\text{TN}+\text{FP}} \end{aligned} \tag{18}$$

As shown in Figure 6, a true positive (TP) result was achieved when a person who is 'politically left' was classified as such, and a false negative (FN) result was obtained when a person who is 'politically left' was classified as 'politically right.' A true negative (TN) result was observed when a person who is 'politically right' was classified as 'politically right,' and a false positive (FP) result was found when a person who is 'politically right' was classified as 'politically left.' When the accuracy was higher, it meant that the classification model performed better.

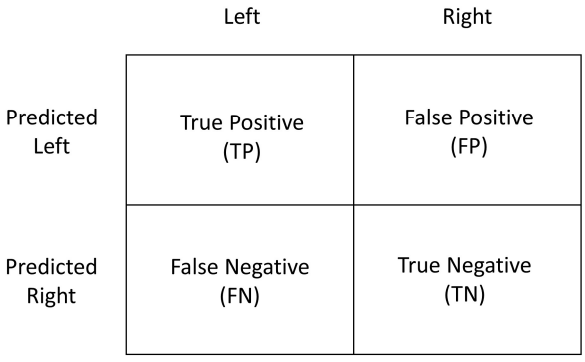

Figure 6. Performance metric.

4.2. News and Comment Data

We collected approximately 100,000 news articles from Daum in April 2017 [7]. Then, we collected approximately 2.68 million comments written by 200,000 users on these articles. We created the database tables News_list and Comments to respectively store news data and comment data.

$$\begin{array}{l} \textit{News\_list (Num, Subject, Post\_ID, Company, News\_Time, News\_Date)} \\ \textit{Comments (Num, ID, Count, Content, Time, Post\_ID, Name, Company)} \end{array} \quad (19)$$

Table 2 lists the top five news articles in terms of number of comments. The top news article had 9754 comments written by 7427 users, indicating that each user wrote 1.3 comments (i.e., more than one) on average.

Table 2. Top 5 news articles in terms of number of comments.

Num	Post_ID	Subject	Company	Num of Comments	Num of Users
1	20170407085950355	Ahn Copies Obama's Speech	Herald	9754	7427
2	20170407163335851	Sewol-ho wasting 0.1 B$	Yonhap	9266	8067
3	20170402160552920	Taegukgi Gathering in Bongha	News 1	7367	6484
4	20170401154147438	Moon, Park Amnesty	Newsys	6687	4917
5	20170404111730189	Stray TK	Edaily	5553	4765

Table 3 lists user data for the top five commenters. The first user wrote 673 comments on 634 news articles in one month; in other words, this user wrote an average of 23 comments each day. Further, the second user wrote 652 comments on 250 news articles; in other words, this user wrote an average of 2.6 comments for each news article.

Table 3. User data for top 5 commenters.

Num	ID	Name	Num of Comments	Num of News Articles
1	9010433	Sleeping User	673	634
2	-135404000	Happycat	652	250
3	16476611	Kwakyongwoo	643	486
4	-144133664	Candy	622	461
5	-118722416	Moonse~	599	276

4.3. Similarty Comparison Using SWR

In the first experiment, we evaluated how the similarity between comments increases when SWR is used. The model computed the average similarity between the first user's comment and other comments using SWR as follows:

$$Sim_{SWR} = \sum_{j=2}^{n} Sim_{SWR}(C_1, C_j)/(n-1) \quad (20)$$

The result is shown in Figure 7. We sampled n comments from the top 500 news commenters. When SWR was not applied, the average similarity score was 0.088. However, when SRW was used, the average similarity score was 0.284. When SWR was used, similarity score increased about 3.2 times compared to when we did not use SWR. It means that we are better at finding similar comments using SWR.

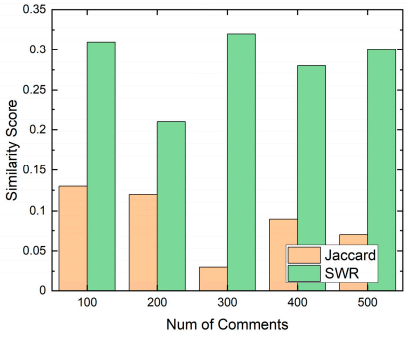

Figure 7. Similarity Score using SWR.

In the following example, there are two comments, C_1 and C_2:

$$\begin{gathered} C_1: \text{칠푼이가 실수는 많이했다. 그래서 사법결정도 없이 정치적책임으로} \\ \text{사임하고 국회에 절차를 줘라 했는데 끝까지 탄핵으로} \\ \text{(Seven pennies made a lot of mistakes. So, she should have resigned due to} \\ \text{political responsibility without making a judical decision and asked the National} \\ \text{Assembly to give the procedure to her, but she ended up being impeached.)} \\ C_2: \text{참 정말 한심하네 앵가히 해라} \\ \text{(It is really pathetic. Do it in moderation.)} \end{gathered} \tag{21}$$

The word sets S_1 and S_2 for comments C_1 and C_2 are as follows:

$$\begin{gathered} S_1 = \{\text{칠푼 (Seven pennies), 실수 (mistakes), 많이했다 (made a lot),} \\ \text{사법결정 (judical decision), 정치적책임 (political responsibility), 사임 (resign),} \\ \text{국회 (National Assembly), 절차 (procedure), 끝 (end), 탄핵 (impeach)}\} \\ S_2 = \{\text{한심 (pathetic), 앵가히 (moderation), 해 (do)}\} \end{gathered} \tag{22}$$

When SWR was not used, the similarity score between C_1 and C_2 was 0 because there was no intersection between S_1 and S_2. However, when SWR was used, the words in S_2 were replaced as follows:

$$\begin{gathered} \text{한심 (pathetic)} \rightarrow \text{끝 (end)} \\ \text{앵가히 (moderation)} \rightarrow \text{끝 (end)} \\ \text{해 (do)} \rightarrow \text{탄핵 (impeach)} \end{gathered} \tag{23}$$

Then, S_2' is computed as follows.

$$S_2' = \{\text{끝 (end), 탄핵 (impeach)}\} \tag{24}$$

Therefore, when SWR is used, the similarity score between C_1 and C_2 is 0.2. When SWR was applied, the two comments were identified as similar.

4.4. Political Stance Classification Using SWR

SWR was used to classify each user's political stance. When a new comment was analyzed that contained a word not trained in the deep learning model, the unknown word was replaced with a similar word using SWR. We then evaluated how much the accuracy increased using five-fold cross-validation [34]. In SWR-*k*, as shown in Figure 8, *k* indicates the number of similar words in SWR. In SWR-1, our model used only one similar word. In SWR-10, our model used ten similar words. Note that even though a word in the comment is similar to another word, if the other word is not in the trained word set, it was not replaced.

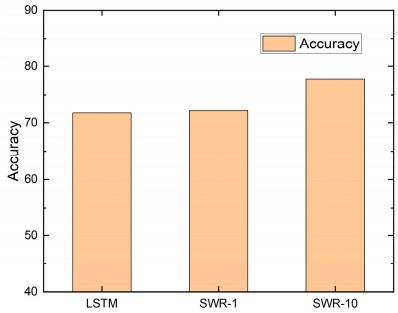

Figure 8. Accuracy.

We trained the seq2seq model using comments written by 400 individuals and tested it using 100 users' comments written about news stories via five-fold cross-validation. When SWR was not used, the accuracy of the seq2seq model was 71.71%. When we used SWR-1, its accuracy increased by 0.4% to 72.11%. When we used SWR-10, the accuracy was 77.77% and increased by 6.06% compared to when SWR was not used. Therefore, we show that the accuracy to classify each user's political stance increases.

When SWR was not used, the recall of the seq2seq model was approximately 86%, as shown in Figure 9. When we used SWR-1, its recall increased by 5% to 91%. When we used SWR-10, the recall was 86%. On the contrary, when SWR was not used, the false positive rate (FPR) of the seq2seq model was approximately 69%, as shown in Figure 10. When we used SWR-1, its FPR decreased by 3% to 66%. When we used SWR-10, the FPR was 41% and decreased by 28% compared to when SWR was not used. Therefore, we show that the FPR to classify each user's political stance decreases.

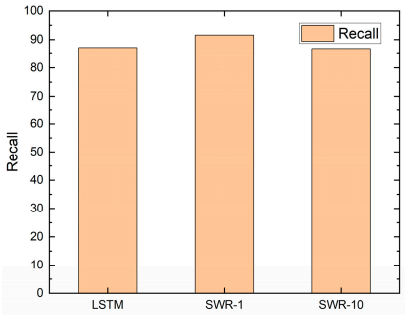

Figure 9. Recall.

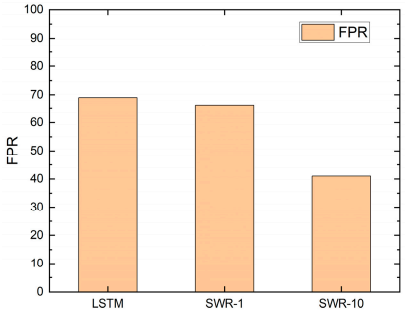

Figure 10. FPR.

An example of SWR's role in political stance classification is as follows. The following comment is labelled as 'politically left.' However, when our model did not use SWR, it was incorrectly classified as 'politically right.' The comment is as follows:

$$C_1: \text{홍준표도 필사즉생을 실천자신의 망발로써 문재인에게 표몰이 해줘서 욕받이시킬 의도가 있었다면 표몰이 효과만 만점자신 (Hong Joon-pyo also practiced the idea of living if he wanted to die. If he had intentions to insult Moon Jae-in by supporting him with his own remarks, he would only have the effect of support)} \quad (25)$$

The word set of this comment is as follows:

$$S_1 = \{\text{홍준표 (Hong Joon-pyo), 필사즉생 (the idea of living if he wanted to die), 실천자신 (practice), 망발 (remarks), 문재인 (Moon Jae-in), 표몰 (support), 욕받이시킬 (insult), 의도 (intentions), 표몰 (support), 효과 (effect), 만점자신 (only have)}\} \quad (26)$$

When our model used SWR, the word set was converted as follows:

$$S_1' = \{\text{홍준표 (Hong Joon-pyo), 홍준표 (Hong Joon-pyo), 홍준표 (Hong Joon-pyo), 홍준표 (Hong Joon-pyo), 문재인 (Moon Jae-in), 문재인 (Moon Jae-in), 문재인 (Moon Jae-in), 한계다메르켈봐 (Limit), 문재인 (Moon Jae-in), 칠푼 (Seven pennies), 칠푼 (Seven pennies)}\} \quad (27)$$

Thereafter, it was classified as 'political left.' Therefore, our model's ability to accurately classify each user's political stance increased when using SWR.

In addition, we evaluated how many similar comments each user writes using SWR. As shown in Table 4, when we use Jaccard, only four users obtain scores between 50 and 60. However, when we use SWR, two more users obtain scores between 50 and 60.

Table 4. Comparison of comment similarity.

	0~10	10~20	20~30	30~40	40~50	50~60	60~70	70~80	80~90	90~100
Jaccard	61	8	7	7	3	4	2	2	1	5
SWR	61	7	7	7	2	6	2	2	1	5
Diff	0	−1	0	0	−1	2	0	0	0	0

On the other hand, when we use SWR, the similarity scores increased compared to Jaccard as shown in Figure 11. The average of the difference is 0.7752. For example, a user obtains a similarity score which is 15 points higher than using Jaccard. Therefore, we think that we can find similar comments better by using SWR.

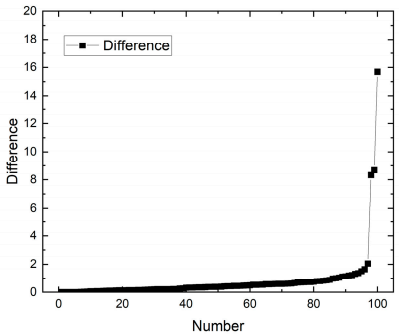

Figure 11. Difference between Jaccard and SWR.

Finally, we provided the political stances of the top users, as depicted in Figure 12. Seven users are classified as 'politically left,' and six users are classified as 'politically right'.

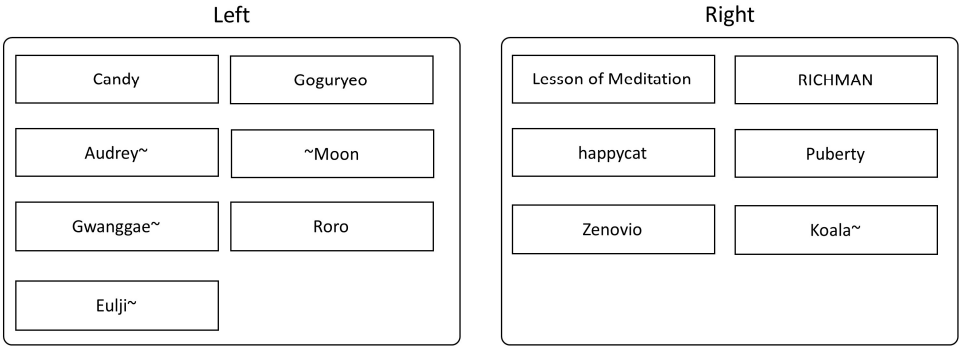

Figure 12. Top users' political stances.

5. Discussion

We performed two experiments to show the effect of SWR. The first one was to increase similarity between two comments by using SWR and second one was to increase the accuracy in classifying each user's political stance. Our model showed that SWR was beneficial in both experiments.

However, when fabricating the similar word database using word2vec, we only used the top 500 users' comments. If we used additional users' comment data, we expect that the effectiveness of SWR would increase because we can find more similar words when there are more words. In addition, a graph neural network [35,36], which can consider the relations between commenters, could increase the accuracy of political stance classification.

In addition, when we used the SWR for political stance classification, the FPR decreased by 28% compared to when we did not use the SWR. However, even though we used the SWR, the FPR was still 41%. We believe that this is because the number of 'politically left' comments is three times larger than the number of 'politically right' comments. In the future, we need to use more 'politically right' comments and obtain a balance between the 'politically left' and 'politically right' comments.

However, we believe that our method can be applied to other research areas. Online shopping markets, such as Amazon [37] and Coupang [38], have increased recently. They allow their customers to write comments about the products they buy from them. However, comments can be either malicious or promotional. We believe that we can reduce malicious or promotional comments using our methods.

6. Conclusions

A number of individuals read news articles on portal sites and affect public opinion by writing similar comments, which has become a significant social problem. In our previous research, we used Jaccard similarity to determine the similarity between two comments. However, when Jaccard similarity was used, the meaning of the comments was not considered. In order to mitigate this issue, we proposed SWR to compute similarity between comments more correctly.

First, we fabricated a similar word database using word2vec and data from the comments on news articles. Second, our trained model used SWR to compute the similarity between comments. We showed that when the model used SWR, its similarity detection performance increased about 3.2 times, and its accuracy when predicting a commenter's political stance increased approximately 6%.

We expect that by using SWR, we can compute the similarity between comments more accurately. In addition, the SWR can be applied to analyze users' comments or opinions in many other areas, such as online shopping.

Author Contributions: Conceptualization, D.L. and S.C.; methodology, S.C.; software, D.L. and S.C.; validation, D.L. and S.C.; writing—original draft preparation, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2019R1G1A11100261) and was supported by research funds for newly appointed professors of Jeonbuk National University in 2021.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. User Ratio Reading News from Portal Sites. Available online: https://www.dailyimpact.co.kr/news/articleView.html?idxno=50488 (accessed on 10 January 2022).
2. Naver. Available online: http://www.naver.com (accessed on 10 January 2022).
3. Daum. Available online: http://www.daum.net (accessed on 10 January 2022).
4. Ji-Hye, J. Assembly's NIS Prove Fizzles Out. *Korea Times*. 20 August 2018. Available online: http://www.koreatimes.co.kr/www/nation/2013/08/113_141397.html (accessed on 10 January 2022).
5. Suh-yoon, L. Governor Kim Kyoung-Soo Sentenced to 2 Years for Online Opinion Rigging. *Korea Times*. 30 January 2019. Available online: http://www.koreatimes.co.kr/www/nation/2019/01/113_262961.html (accessed on 10 January 2022).
6. Shin, H. Kraken to Detect Malicious Comments. *JoongAng*. 30 December 2021. Available online: https://www.joongang.co.kr/article/25036975#home (accessed on 10 January 2022).
7. Choi, S. Internet News User Analysis Using Deep Learning and Similarity Comparison. *Electronics* **2022**, *11*, 569. [CrossRef]
8. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**, arXiv:1409.3215.
9. Jaccard Index. Available online: https://deepai.org/machine-learning-glossary-and-terms/jaccard-index (accessed on 10 January 2022).
10. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
11. Roh Moo-hyun. Available online: https://en.wikipedia.org/wiki/Roh_Moo-hyun (accessed on 14 April 2022).
12. Hamborg, F.; Donnay, K. NewsMTSC: A Dataset for (multi-)Target-dependent Sentiment Classification in Political News Articles. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, Online, 19 April 2021.
13. Wikipedia. Available online: https://en.wikipedia.org/wiki/Main_Page (accessed on 10 January 2022).
14. Recasens, M.; Danescu-Niculescu-Mizil, C.; Jurafsky, D. Linguistic Models for Analyzing and Detecting Biased Language. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; pp. 1650–1659.
15. Hube, C.; Fetahu, B. Detecting Biased Statements in Wikipedia. In Proceedings of the World Wide Web Conference, Lyon, France, 23–27 April 2018.
16. Fan, L.; White, M.; Sharma, E.; Su, R.; Choubey, P.K.; Huang, R.; Wang, L. In Plain Sight: Media Bias through the Lens of Factual Reporting. In Proceedings of the 2019 Conference on Empirical Methods in National Language Processing, Hong Kong, China, 3–7 November 2019; pp. 6343–6349.
17. Cho, D.B.; Lee, H.Y.; Jung, W.S.; Kang, S.S. Automatic Classification and Vocabulary Analysis of Political Bias in News Articles by Using Subword Tokenization. *KIPS Trans. Softw. Data Eng.* **2021**, *10*, 1–8.
18. Garrett, R.K. Echo chambers online? Politically motivated selective exposure among Internet news users. *J. Comput.-Mediat. Commun.* **2009**, *14*, 265–285. [CrossRef]
19. Latane, B. The psychology of social impact. *Am. Psychol.* **1981**, *36*, 343–356. [CrossRef]
20. Social Impact Theory. Available online: https://en.wikipedia.org/wiki/Social_impact_theory (accessed on 17 June 2022).
21. Social Inertia. Available online: https://en.wikipedia.org/wiki/Social_inertia (accessed on 17 June 2022).
22. Duradoni, M.; Gronchi, G.; Bocchi, L.; Guazzini, A. Reputation matters the most: The reputation inertia effect. *Hum. Behav. Emerg. Tech.* **2020**, *2*, 71–81. [CrossRef]
23. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
24. GloVe. Available online: https://en.wikipedia.org/wiki/GloVe (accessed on 17 June 2022).
25. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *arXiv* **2016**, arXiv:1607.04606. [CrossRef]
26. fastText. Available online: https://en.wikipedia.org/wiki/FastText (accessed on 17 June 2022).

27. Sitaula, C.; Basnet, A.; Mainali, A.; Shahi, T.B. Deep Learning-Based Methods for Sentiment Analysis on Nepali COVID-19-Related Tweets. *Comput. Intell. Neurosci.* **2021**, *2021*, 2158184. [CrossRef] [PubMed]
28. Shahi, T.B.; Sitaula, C.; Paudel, N. A Hybrid Feature Extraction Method for Nepali COVID-19-Related Tweets Classification. *Comput. Intell. Neurosci.* **2022**, *2022*, 5681574. [CrossRef] [PubMed]
29. Cosine Similarity. Available online: https://en.wikipedia.org/wiki/Cosine_similarity (accessed on 14 April 2022).
30. Olah, C. Understanding LSTM Networks. Available online: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 10 January 2022).
31. Hannanum. Available online: https://konlpy-ko.readthedocs.io/ko/v0.4.3/api/konlpy.tag/ (accessed on 26 April 2022).
32. Keras. Available online: https://keras.io/ (accessed on 26 April 2022).
33. Beautifulsoup4. Available online: https://pypi.org/project/beautifulsoup4/ (accessed on 26 April 2022).
34. Cross Validation. Available online: https://en.wikipedia.org/wiki/Cross-validation_(statistics) (accessed on 26 April 2022).
35. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
36. Choi, S. Malicious Powershell Detection using Graph Convolution Network. *Appl. Sci.* **2021**, *11*, 6429. [CrossRef]
37. Amazon. Available online: https://www.amazon.com (accessed on 16 June 2022).
38. Coupang. Available online: https://www.coupang.com (accessed on 16 June 2022).