Hindawi
Complexity
Volume 2018, Article ID 7408431, 9 pages
https://doi.org/10.1155/2018/7408431

WILEY | Hindawi

Research Article

Using Deep Learning to Predict Sentiments: Case Study in Tourism

C. A. Martín, J. M. Torres, R. M. Aguilar, and S. Diaz

Department of Computer and Systems Engineering, Universidad de La Laguna, 38200 La Laguna, Tenerife, Spain

Correspondence should be addressed to R. M. Aguilar; raguilar@ull.edu.es

Received 19 April 2018; Accepted 23 September 2018; Published 23 October 2018

Guest Editor: Ireneusz Czarnowski

Technology and the Internet have changed how travel is booked, the relationship between travelers and the tourism industry, and how tourists share their travel experiences. As a result of this multiplicity of options, mass tourism markets have been dispersing. But the global demand has not fallen; quite the contrary, it has increased. Another important factor, the digital transformation, is taking hold to reach new client profiles, especially the so-called third generation of tourism consumers, digital natives who only understand the world through their online presence and who make the most of every one of its advantages. In this context, the digital platforms where users publish their impressions of tourism experiences are starting to carry more weight than the corporate content created by companies and brands. In this paper, we propose using different deep-learning techniques and architectures to solve the problem of classifying the comments that tourists publish online and that new tourists use to decide how best to plan their trip. Specifically, in this paper, we propose a classifier to determine the sentiments reflected on the http://booking.com and http://tripadvisor.com platforms for the service received in hotels. We develop and compare various classifiers based on convolutional neural networks (CNN) and long short-term memory networks (LSTM). These classifiers were trained and validated with data from hotels located on the island of Tenerife. An analysis of our findings shows that the most accurate and robust estimators are those based on LSTM recurrent neural networks.

1. Introduction

That the world of tourism is changing is not news. There are more and more data, both structured and nonstructured, being generated at ever higher rates, which once transformed into information, which provide a tangible value to businesses. Opinion mining and sentiment analysis is a very active field of study in recent years [1, 2]. The problem is determining how to benefit from these data and how to use them to generate value. Although future data is out of reach, it is possible to predict what will happen based on past data through a process known as predictive analytics.

The use of automatic tools in social networks for the tourism sector has generated ample literature due to the importance of influencing the consumer's participation and affecting the way in which tourists perceive their experience [3]. Thus, for a tourism company to grow, it is essential that it crafts forecasts to optimize its marketing campaigns and the performance of its corporate website in order to improve the feedback from its users. The predictive scores obtained from the models for each client indicate the steps that should be taken to achieve the goals of retaining the client, upselling a product, or offering him a new service. Reliable and consistent information is needed in order to forecast client behavior. The work presented in this paper builds on this idea, testing methods to analyze the reviews clients provide on digital platforms concerning the service they received.

In contrast to the traditional conversations that take place in specific physical locations, the digital conversation is shaped using new methods and tools for engaging the public, whose social interaction characteristic is the focus of its dynamic [4, 5]. Basically, it is the so-called electronic word of mouth (eWOM), the features of which lead to completely different consequences [6] from traditional conversations. These features focus on the ease with which the message is distributed via social media, which are online communications platforms where users create their own content using Web 2.0 technologies that enable the writing, publication,

and exchange of information (Kaplan and Haenlein, 2010). The incorporation of eWOM into social media or the Web 2.0 has the following properties:

(i) Great ability to disseminate, where users can access opinions from strangers

(ii) Massive engagement by users of different ages and groups, all sharing different points of view

(iii) The message spreading quickly in several ways: blogs, websites, social media, messages posted in online groups, etc.

(iv) Multidirectional discussion among users, who play an active role by answering questions on the information presented

(v) Persistence over time, since the discussions are uploaded for the current and future reference

(vi) Credibility, since the information is offered by users spontaneously and, in theory, with no profit motive

These features make monitoring of eWOM by tourism companies particularly relevant [7].

According to [7], an eWOM is "any positive or negative statement made by potential, current, or former customers about a product or company that will be made available to a large number of people and institutions through the Internet." The main goal of this paper is to classify the positive and negative statements made by tourists by using various deep-learning techniques.

The tourism industry is of great importance in the Canary Islands, as it is the real engine of growth and development in the archipelago, accounting for a high percentage of its GDP. This has a knock-on effect on the remaining industries and services in the islands, especially in the development of trade, transportation, food production, and industry. Tourism also comprises a very important component in creating jobs in the service sector of the archipelago, which encompasses direct employment in the sun and beach sector, as well as workers in activities that support tourism, such as restaurants, hotels, travel agencies, passenger transport, car rental, and recreational, cultural, and sports activities.

In 2016, Tenerife, one of the Canary Islands, received over seven million tourists, most of them from the United Kingdom, which accounted for thirty-one percent of the passengers arriving at Tenerife's airports in 2016. These figures indicate that the opinion that English tourists have of Tenerife is particularly relevant. As a result, we will conduct our experiments based on the comments in English made about the hotels in Tenerife.

2. Materials and Methods

Below, we describe the estimators implemented, the data used, and the procedures employed to train, evaluate, and compare the estimators.

2.1. Data Sets. The techniques used, which will be detailed in the sections that follow, fall under the category of supervised learning methods. This requires having a set of previously classified data before the prediction system can be trained and a test sample in order to validate how accurately the technique behaves. To satisfy this requirement and in order to use real data in the methods, we extracted information from reviews in English on the http://booking.com and http://tripadvisor.com websites for hotels on the island of Tenerife. The structure of the information varies depending on the portal that is used as the source of data:

(i) Booking

  (a) The comment will have a general score between 0 and 10

  (b) The comment's author is able to separate positive and negative aspects

(ii) TripAdvisor

  (a) The comment features a rating system using bubbles or stars to assign a score between 1 and 5

  (b) There is a single field for expressing an opinion

In order to extract the information, we developed Python scripts based on the Scrapy framework and adapted them to the domain using the scripts offered by the MonkeyLearn project [8].

As a result of this process, we obtained more than 40,000 records with different fields, depending on the source portal for the data: title, comment, score, date, and location of the visitor.

The preparation of the data set so that it can be used in the deep-learning techniques studied in this paper is an important part of the work. In the initial phase and in an effort to standardize the information taken from the two sources used, we developed programs in Python to build a CSV file with two columns:

(i) Comment (free text)

(ii) Label ("Bad" or "Good")

Depending on the portal from which the data were sourced, the scripts had different functionalities.

In the case of the samples taken from TripAdvisor, the original title and comment were used to comprise a single text containing the visitor's full opinion. Those reviews scoring three or higher were labeled "Good," and those scoring two or below were labeled "Bad." Any samples with an intermediate rating were discarded so as not to hamper the training.

In the case of the samples taken from Booking, we evaluated the number score awarded by the visitor. For scores higher than six, which were labeled "Good," the comment was generated by concatenating the title and comment fields. For scores of four and below, the title and negative comment were concatenated and labeled "Bad."

Once the structure of the samples was standardized, the data set was divided into three parts. For the first set, a random balanced selection (half of the samples labeled "Good"

Figure 2: Positive review words.

Figure 1: Negative review words.

and the other half labeled "Bad") of 9640 samples was carried out. The second set was used to evaluate the models after each training epoch, and finally, the test sample was created using 2785 samples in which the scores assigned by the tourists were known and which we used to compare the accuracy of the various models.

The second phase to prepare the data sets for the deep-learning models involved adapting them to the data domain that can be input into the models. Each comment used from the training set was subject to preprocessing before it could be used. Fortunately, Python offers an ecosystem of libraries that can be used in several machine learning applications [9] like the Python NLTK library [10] that allow us to perform this task. The comments were processed as follows: separation into words, deletion of punctuation and any nonalphabetic terms, and lastly deletion of words identified as stop words, which provide no information when determining if a comment is positive or negative.

The figures below offer a visual representation of those words that appeared most frequently in the positive and negative comments:

As we can see by analyzing Figures 1 and 2, there are certain words that clearly identify a polarized sentiment like "Good" or "Bad." However, this does not apply to many other words, which are sometimes even included in comments with the exact opposite meaning (note that the group of words used in negative comments contains words like "better," "nice," or "clean"). As a result, a simple classification of the comment based on the appearance or absence of certain words is not sufficient; rather, machine-learning techniques, such as the ones used in this paper, are needed to analyze the relationships between the words.

The algorithms used require as an input a fixed-length vector in which each component is a number. In the technique for coding text into number vectors known as bag of words (BoW), a dictionary is created with the words found most frequently in all of the training comments. Each comment is then coded into a fixed-length vector corresponding to the number of words in the dictionary created. In BoW, a comment is coded into a vector in which each component counts how many times each word in the dictionary appears in the comment. We ruled out this coding method because even though it represents the frequency of words in the comment, it discards information involving the order in which those words appear in the comment.

The word embedding technique is currently one of the best for representing texts as number vectors. It is a learned representation in which words with similar meanings are given a similar representation. Each vocabulary word is represented with a vector, and its representation is learned based on the use it is given in the training comments. As a result, words that are used in a similar way will have a similar representation. The learning process for embedding is carried out in this paper by adding a layer at the front of the neural network in each of the models. In order to be able to generate our models in Python, we resorted to the Keras library [11].

In order to use this library and add the embedding layer to our models, we have to first transform the tourists' comments into integer vectors in which each word is represented by an index in a list of words. We do so by using the Tokenizer class in Keras, creating an instance based on the training data set and limiting the size of the vocabulary, in our case, to the 5000 most common words. By using the Tokenizer instance, we transform each comment into a vector of variable length in which each word is an integer where the value i corresponds to the i-th most used word in the samples. Since the tourist comments do not all have the same length, the comment vectors are filled in with zeros until the specified fixed length is reached that matches the number of words in the longest comment (582 in this work). Figure 3 shows this process.

The embedding layer is initialized with random weights and trained at the same time as the rest of the models, with the training supplied by the training data set. In every model implemented for this work, the embedding layer was used as the first layer in the model, with the following characteristics:

(i) Input dim (the maximum integer value of the vector component input): its value, based on how we coded the comments, is 5000

(ii) Output dim (the length of the vectors that will represent the words after embedding): in most of the experiments, this length is set to 300

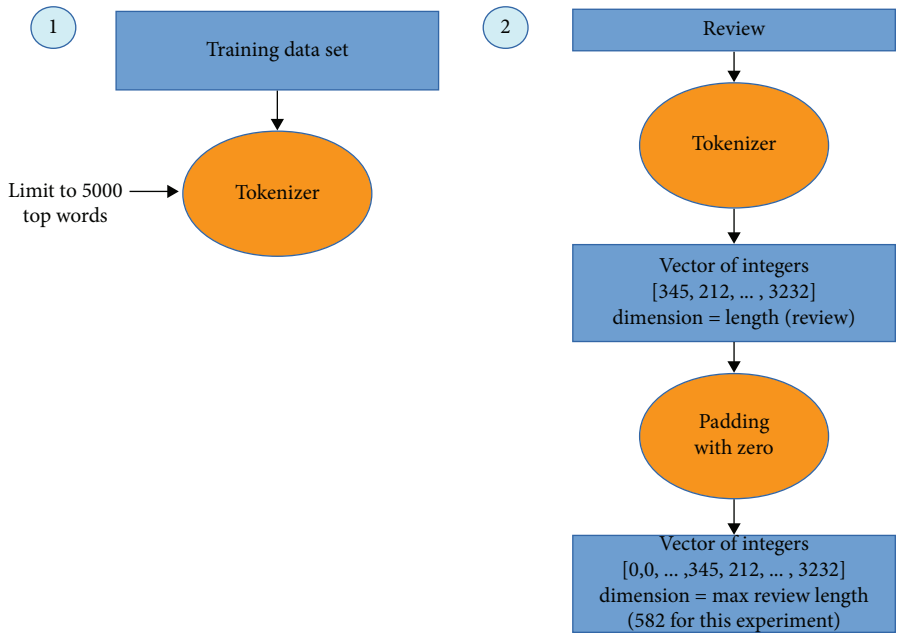

Figure 3: Converting review to fixed-length vectors.

(iii) Input length (the length of the vectors in the layer): As defined earlier, it is the maximum length of the comments in words. In the experiments conducted for this work, the maximum comment length was 582 words

As shown in Figure 4, the output of the embedding layer is a 582×300 matrix in which each comment is transformed into a vector with 582 components, in which each component (representing one word) is coded into a vector of length 300.

2.2. Recurrent Neural Networks. To predict the sentiment of the comments, we use models based on neural networks. Each comment is a sequence of encoded words that can be processed as a time series. However, the most common neural networks (e.g., feed-forward neural networks) lack the memory to store information over time. Recurrent neural networks [12] (RNN) solve this problem by making the network output y_j at step j depend on previous computations through a hidden state s_j that acts as a memory for the network.

Figure 5 shows the RNN we used, unfolded into a full network. By unfolded, we simply mean that we write out the network for a complete sequence of N steps, where x_j is the j-th encoded word in the comment, which we used as the input to the network in the j-th step.

In a RNN, the relationship between output y_j, input x_j, and state s_j in step j is determined by the type of RNN cell. As Figure 5 shows, we used a kind of cell called long short-term memory [13] (LSTM). LSTM recurrent neural networks are capable of learning and remembering over long input sequences and tend to work very well for labeling sequences of word problems [14].

As (1) shows, output y_j depends on the state s_j of the LSTM cell through the activation function $\sigma_y(x)$ (which is generally $\tanh(x)$). The state s_j depends on the state of the previous step s_{j-1} and on the candidate for the new value of the state $\tilde{s}_j$. The output gate o_j controls the extent to which the state s_j is used to compute the output y_j by means of the Hadamard product $(\circ)$. The input gate i_j controls the extent to which $\tilde{s}_j$ flows into the memory, and the forget gate f_j controls the extent to which s_{j-1} remains in memory.

The o_j, f_j, and i_j gates and the candidate for the new value of the state of the cell $\tilde{s}_j$ can be interpreted as the outputs of conventional artificial neurons whose inputs are the input to cell x_j at step j and the output of cell y_{j-1} at $j-1$. The activation function for the gates σ_g is usually the sigmoid function, while for σ_s it is usually $\tanh(x)$.

$$\begin{aligned} f_j &= \sigma_g\left(W_f x_j + U_f y_{j-1} + b_f\right), \\ i_j &= \sigma_g\left(W_i x_j + U_i y_{j-1} + b_i\right), \\ o_j &= \sigma_g\left(W_o x_j + U_o y_{j-1} + b_o\right), \\ \tilde{s}_j &= \sigma_s\left(W_s x_j + U_s y_{j-1} + b_s\right), \\ s_j &= f_j \circ s_{j-1} + i_j \circ \tilde{s}_j, \\ y_j &= o_j \circ \sigma_y\left(s_j\right). \end{aligned} \tag{1}$$

As Figure 5 shows, each input x_j in (1) is the coded value of the successively coded sequence of words $(x_j)_{j=0}^{N-1}$ in the comment. The RNN cell provides an output at each step j,

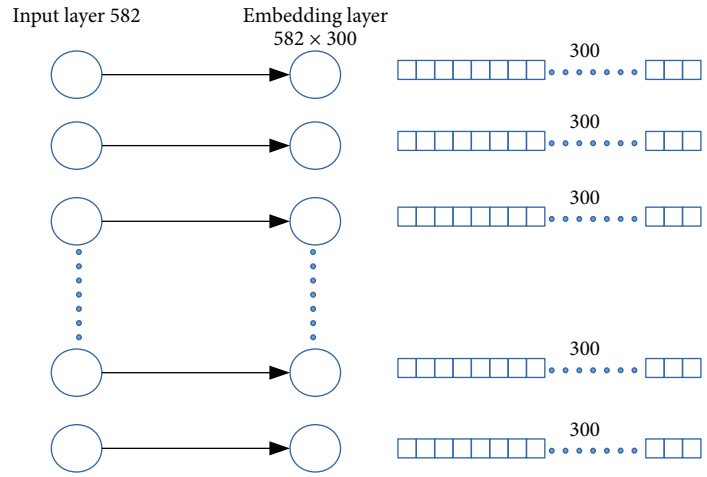

Figure 4: Embedding layer.

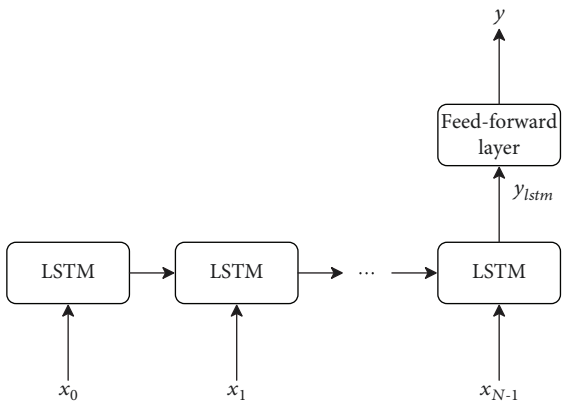

Figure 5: LSTM unfolded into a full network.

but for the prediction, only output y_{N-1} of the network is considered when the last word x_{N-1} is input into the network. This output is what we refer to as y_{lstm} in Figure 5.

Output y_{lstm} is used as an input to a one-neuron feed-forward layer with a sigmoid activation function, the output of which, between 0 and 1, is the network's prediction for whether the sentiment is positive or negative. The output y of that single neuron can be expressed as indicated in

$$y = \sigma_{\text{sigmoid}}\left(\sum_k w_k y_{\text{lstm}} + b\right)\sigma_{\text{sigmoid}}(x) = \frac{e^x}{e^x + 1}, \quad (2)$$

where w_k is used to denote the weight of the k-th input, b is the bias, and $\sigma_{\text{sigmoid}}(x)$ is the sigmoid activation function of the output neuron in the layer.

2.3. Convolutional Neural Networks. Another type of neural network that can be used to predict time series is a convolutional neural network (CNN). These are biologically inspired variants of feed-forward neural networks used primarily in computer vision problems [15], although their ability to exploit spatially local correlation in images can also be used in time-series problems, like sentiment analysis.

In these models, the output of each neuron a_j^l is generated from the output of a subset of spatially adjacent neurons. Every neuron in the same layer shares the same weight and bias, meaning the layer can be expressed in terms of a filter that is convoluted with the output of the previous layer. Equation (3) shows the output a_j^{lk} of the j-th neuron of the l-th convolutional layer:

$$a_j^{lk} = \sigma^{lk}\left(\left(W^{lk*}a^{l-1}\right)_j + b^{lk}\right), \quad (3)$$

where $\left(W^{lk*}a^{l-1}\right)_j$ is the j-th element resulting from the convolution of the filter defined by W^{lk} with the output of the previous layer a^{l-1}, $\sigma^{lk}(x)$ is the activation function for the convolutional layer, and $k \in [0, \ldots, K]$ indicates that it is the output of the k-th channel of the layer. In each convolutional layer, different filters can be applied to the output of the previous layer to generate different representations or channels k, thus yielding a fuller representation of the data.

For the $\sigma^{lk}(x)$ activation function, we used the ReLU function because it does not suffer from the vanishing gradient problem [7] when training the neural network. Equation (4) shows the expression for the activation function.

$$\sigma^{lk}(x) = \sigma_{\text{relu}}(x) = \max(0, x). \quad (4)$$

Figure 6 shows a general diagram of the CNN developed in this paper to analyze sentiment. In order to apply a CNN to a time series, we arranged the encoded words in the same order as they appear in the comment, such that adjacent words in the comment are spatially adjacent at the input to the neural network. Moreover, each word embedding dimension is a different input channel to the network. In this way, the convolutional layers can exploit the local correlation between words in each comment.

After each convolutional layer with a ReLU, activation function is a max-pooling layer, which partitions the input into a set of nonoverlapping ranges and, for each range,

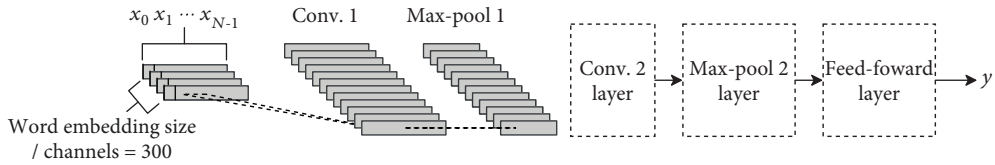

Figure 6: General diagram of the CNNs used.

Table 1: Model structure.

Model	Model description
1	(582) embedding→(582 × 300) LSTM→(30) dense [sigmoid]→(1)
2	(582) embedding→(582 × 300) LSTM→(50) dense [sigmoid]→(1)
3	(582) embedding→(582 × 300) LSTM→(70) dense [sigmoid]→(1)
4	(582) embedding→(582 × 300) LSTM→(100) dense [sigmoid]→(1)
5	(582) embedding→(582 × 300) LSTM→(200) dense [sigmoid]→(1)
6	(582) embedding→(582 × 300) LSTM→(300) dense [sigmoid]→(1)
7	(582) embedding→(582 × 300) LSTM→(500) dense [sigmoid]→(1)
8	(582) embedding→(582 × 300) Conv1D→(575 × 64) MaxPooling1D→(287 × 64) flatten→(18,368) dense [relu]→(10) dense [sigmoid]→(1)
9	(582) embedding→(582 × 300) Conv1D→(575 × 128) MaxPooling1D→(287 × 128) flatten→(36,736) dense [relu]→(10) dense [sigmoid]→(1)
10	(582) embedding→(582 × 300) Conv1D→(575 × 32) MaxPooling1D→(287 × 32) Conv1D→(280 × 64) flatten→(17,920) dense [relu]→(10) dense [sigmoid]→(1)
11	(582) embedding→(582 × 300) Conv1D→(582 × 32) MaxPooling1D→(291 × 32) LSTM→(100) dense [sigmoid]→(1)

Table 2: Training data set.

Training data set	
Positive reviews	4820
Negative reviews	4820
Total reviews	9640
Mean review length (chars)	53
Max review length (chars)	582

outputs the maximum value. Following the convolutional and max-pooling layers is a feedforward layer (as described in (2)) to yield the output of the entire network.

3. Results and Discussion

In order to compare some of the deep-learning techniques mentioned in this paper, we conducted a series of experiments on different models based on LSTM neural networks and CNN. Table 1 shows the structure of each of the models used with the lengths of the inputs at each layer. The first column will be used to refer to the models in subsequent references.

Table 3: Test data set.

Test data set	
Positive reviews	1408
Negative reviews	1377
Total reviews	2785

Table 4: Test results.

Model N	Training time	Good hits	Bad hits	False good	False bad	Accuracy end
1	2208	1227	1228	149	181	88.15
2	2734	1211	1239	138	197	87.97
3	4658	1215	1237	140	193	88.04
4	4406	1198	1261	116	210	88.29
5	4388	1213	1256	121	195	88.65
6	10,630	1221	1263	114	187	89.19
7	13,574	1190	1261	218	116	88.01
8	6994	1210	1247	130	198	88.22
9	9139	1206	1235	142	202	87.65
10	1765	1166	1268	109	242	87.40
11	1602	1187	1272	105	221	88.29

First, we prepared the data as explained in Section 2.1 + 0.1667 emA. The same comment coding technique was used for each test, which included training an embedding layer.

To make the models comparable, we used the same training data set (Table 2) with a total of 9640 reviews. The classification was conducted independently using another data set containing 2785 reviews that were not used during the training. As Table 3 shows, a test data set was employed that was fully balanced between positive and negative reviews.

So as not to use a different number of training epochs based on the model, we decided to use a fixed number, 10, since, for every model, the loss value did not improve significantly with longer training.

Table 4, containing the general results, shows the number of correctly predicted positive and negative reviews, as well as the false positives and negatives.

Models 1 to 6 have the same structure, the number of memory units varying as shown in Table 1. As we can see in Figure 7, the results improve as the number of memory cells is increased, reaching a maximum accuracy of 89.19% in model 6.

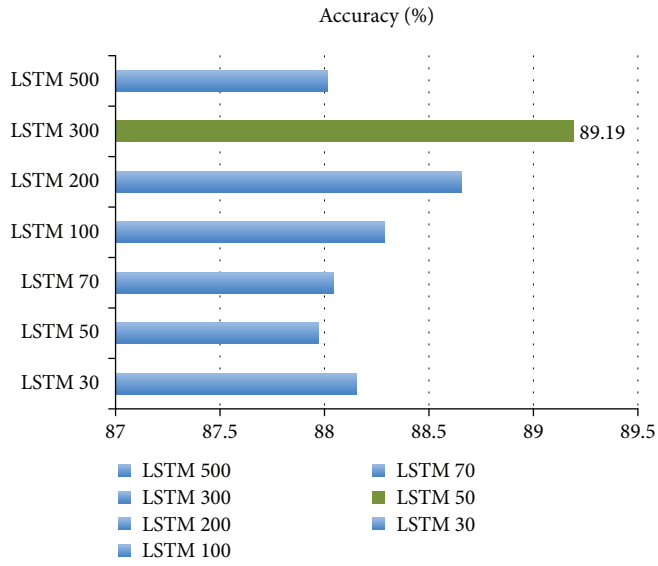

Figure 7: LSTM comparison.

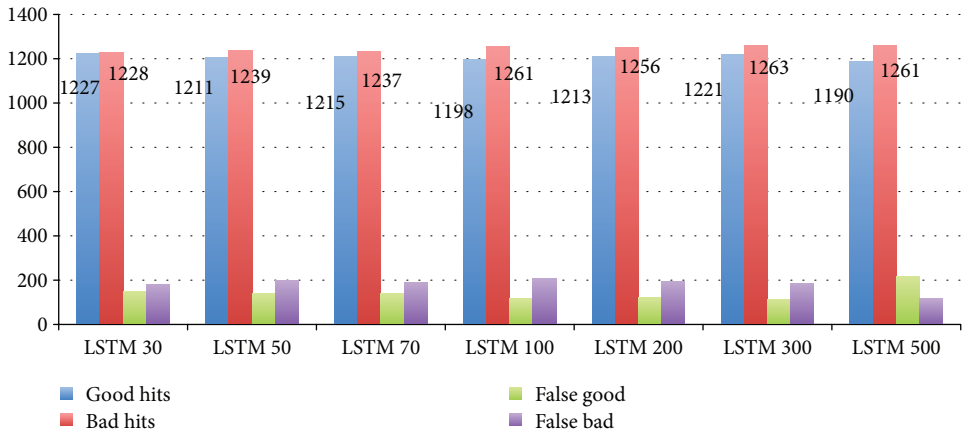

Figure 8: LSTM comparison.

Figure 8 shows the results of the test separated by hits and false positives. The model with the best result improved on the predictions from previous models for negative reviews.

The second part of the experiment consisted of checking for a significant variation when the number of filters was changed in the CNN models. Specifically, we compared models 8 and 9 with one another, yielding the results shown in Figure 9. As we can see, the difference is not significant, and increasing the number of filters does not provide any improvement.

To complete the study, we compared the results for the previous models that yielded the best outcome (model 6 LSTM and model 8 CNN) with models 10 (two-layer CNN) and 11 (CNN and LSTM). As Figure 10 shows, the LSTM models exhibit the highest accuracy, a result that is not improved by adding a convolutional layer.

Figure 11 shows that the best overall result for the LSTM neural networks is based primarily on their better prediction of positive results, in comparison to the other networks trained.

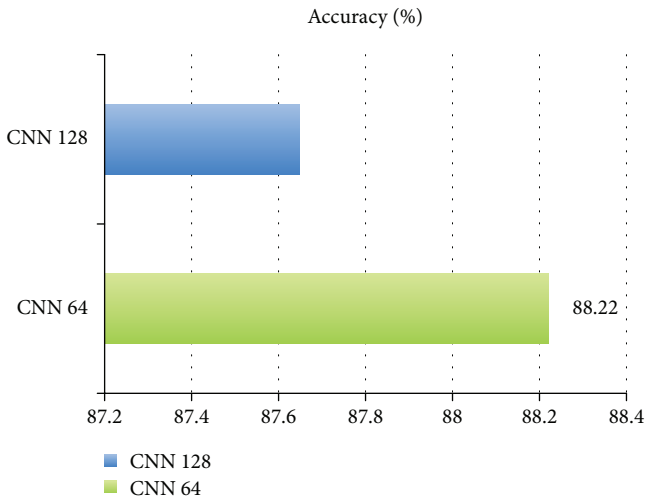

Figure 9: CNN comparison.

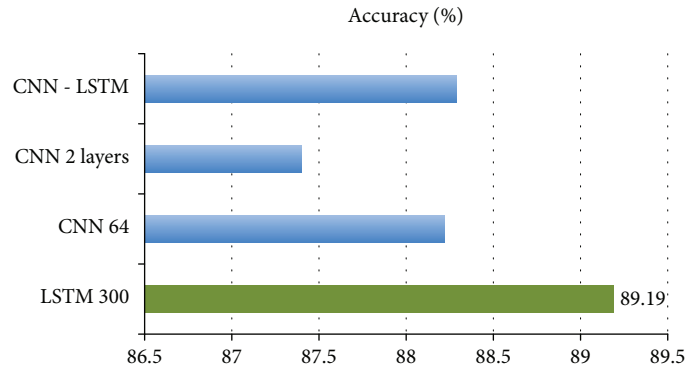

Figure 10: Comparison of LSTM and CNN models.

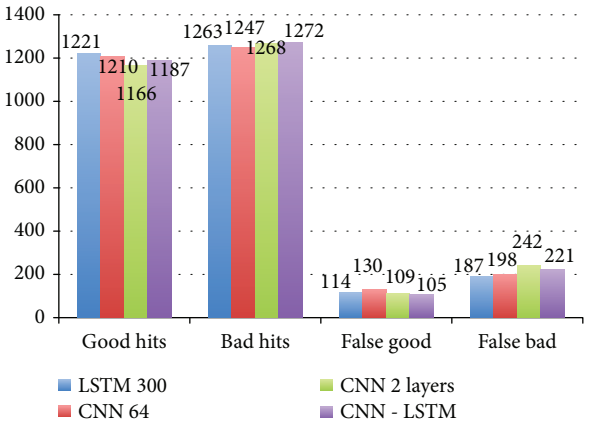

Figure 11: LSTM comparison.

4. Conclusions

In this paper, we considered the problem of predicting sentiment in tourist reviews taken from eWOM platforms for hotels at an important international tourist destination. The use of techniques and automatic tools such as those considered in this paper are very useful for tourism industry practitioners [3]. Specifically, the prediction of positive or negative sentiment expressed by a tourist can be used in different marketing and service management applications:

(i) Comparison of feeling about another local competitor or tourist destination (market positioning)

(ii) Performing a proactive customer service management, generating a job ticket when a negative review is detected (customer management)

(iii) As a measure of indicators to start a campaign to improve the reputation (marketing management)

(iv) As a measure of risk indicators that affect the hotel or destination image (risk management)

Once the models studied in this article have been trained, they can be used in combination with other tools (review extraction or dashboards).

We used deep-learning techniques to devise different predictors based on neural networks, which were trained with the extracted data to compare the accuracies of each.

The predictors evaluated were based on recursive neural networks with cell LSTM and convolutional neural networks. Different designs were considered for each. The methodology was checked by training and validating the model with samples taken from Booking and TripAdvisor.

The results show that LSTM neural networks outperform CNN. The optimum result for CNN is attained with a single convolutional layer and 64 channels. More layers or more channels result in symptoms of overfitting. The LSTM neural networks yield higher accuracies, with one LSTM with a vector length of 300 for the internal state yielding an accuracy just over 89%, the highest for any model.

Finally, the results show that the better results of the neural networks are due primarily to their advantage when classifying the positive comments. They also show that combining convolutional layers with recurrent LSTM layers does not yield any advantages.

Data Availability

The comment tourist review data used to support the findings of this study have been deposited in the GitHub repository https://github.com/camartinULL/Deep_Learning_Predict_Sentiments.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is sponsored by the "VITUIN: Vigilancia Turística Inteligente de Tenerife en Redes Sociales" project, through research funds of the Fundación CajaCanarias.

References

[1] R. Piryani, D. Madhavi, and V. K. Singh, "Analytical mapping of opinion mining and sentiment analysis research during 2000–2015," *Information Processing and Management*, vol. 53, no. 1, pp. 122–150, 2017.

[2] J. A. Balazs and J. D. Velásquez, "Opinion mining and information fusion: a survey," *Information Fusion*, vol. 27, pp. 95–110, 2016.

[3] M. D. Sotiriadis, "Sharing tourism experiences in social media: a literature review and a set of suggested business strategies," *International Journal of Contemporary Hospitality Management*, vol. 29, no. 1, pp. 179–225, 2017.

[4] M. D. Sotiriadis and C. van Zyl, "Electronic word-of-mouth and online reviews in tourism services: the use of twitter by tourists," *Electronic Commerce Research*, vol. 13, no. 1, pp. 103–124, 2013.

[5] R. A. King, P. Racherla, and V. D. Bush, "What we know and don't know about online word-of-mouth: a review and synthesis of the literature," *Journal of Interactive Marketing*, vol. 28, no. 3, pp. 167–183, 2014.

[6] L. de Vries, S. Gensler, and P. S. H. Leeflang, "Popularity of brand posts on brand fan pages: an investigation of the effects of social media marketing," *Journal of Interactive Marketing*, vol. 26, no. 2, pp. 83–91, 2012.

[7] T. Hennig-Thurau, K. P. Gwinner, G. Walsh, and D. D. Gremler, "Electronic word-of-mouth via consumer-opinion platforms: what motivates consumers to articulate themselves on the Internet?," *Journal of Interactive Marketing*, vol. 18, no. 1, pp. 38–52, 2004.

[8] "Hotel review analysis. monkey learn," 2018, https://github.com/monkeylearn/hotel-review-analysis.

[9] R. M. Aguilar Chinea, J. M. Torres Jorge, and C. A. M. Galán, "Aprendizaje automático en la identificación de sistemas. un caso de estudio en la generación de un parque eólico," *Revista Iberoamericana de Automática e Informática industrial*, vol. 16, 2018.

[10] "NLTK python library," 2018, https://www.nltk.org.

[11] "Keras python library," 2018, https://keras.io.

[12] J. Schmidhuber, "Learning complex, extended sequences using the principle of history compression," *Neural Computation*, vol. 4, no. 2, pp. 234–242, 1992.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] A. Graves, *Supervised sequence labelling with recurrent neural networks*, Springer, 2012.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010*, Chia Laguna Resort, Sardinia, Italy, 2010.