Keyword Extraction from Online Product Reviews Based on Bi-directional LSTM Recurrent Neural Network

Y. Wang[1], J. Zhang[2]
[1]Department of Supply Chain and Information Management, Hang Seng Management College, Hong Kong
[2]School of Computer Science and Network Security, Dongguan University of Technology, Dongguan, China
(yuewang@hsmc.edu.hk)

Abstract - **Online reviews are acknowledged as an important source of product information when customers make purchasing decisions. However, in the era of information overload, product review data on the Internet are too abundant and contain much irrelevant information. This makes it difficult for customers to find useful reviews. To solve this issue, some e-commerce websites provide keywords for product reviews, but these are generated beforehand and have the potential to distort customers' opinions of products. This paper presents an automatic keyword extraction method based on a bi-directional long short-memory (LSTM) recurrent neural network (RNN). The results of experiments conducted on product reviews obtain by data-crawling jd.com show that the proposed approach has a very high accuracy of keyword extraction. This can help reduce human annotation efforts in e-commerce.**

Keywords - **Deep learning, e-commerce, product design**

I. INTRODUCTION

E-commerce has transformed the way people buy and sell goods and services. Because of technological development, the Internet provides a convenient way for people to make purchases without having to visit actual stores. An online store can reach customers around the world. According to Google, 51% of Americans prefer online shopping over shopping in stores, and the intention to shop online among Hong Kong residents increased from 75.5% in 2011 to 82 % in 2015.

Currently, almost all e-commerce platforms enable customers to input their reviews of purchased products. With the increase in e-commerce transactions, product reviews and comment data are surging in e-commerce websites. The review data contain consumer sentiment toward or opinions of purchased products or services. These have been a valuable resource for companies to elicit customers' opinions, analyze user behavior and conduct market analysis. They also make it possible for companies to better understand the market, which allows e-commerce companies to distinguish themselves from their competitors, and to accurately predict market demand, facilitating future product development. In addition, many potential product buyers check product review data to better understand products [1]. According to a survey conducted by Deloitte's Consumer Products Group, 67% of Internet users browse online product reviews before purchasing products, and 82% think that online product reviews influence their purchasing decisions. Thus, online product reviews serve as the new form of word of mouth.

In the era of big data, there is an enormous amount of product review data online. Reading product reviews can be time consuming and tedious. In this situation, a potential customer may feel overwhelmed, confused and perplexed. It is difficult for him or her to know which review data are useful, a phenomena often referred to as "information overload." To resolve this issue, some e-commerce platforms summarize the review data and provide a user-friendly way for customers to obtain opinions about the product. For example, Figure 1 shows the screenshot of the Huawei Mate 9 product review page on the Tmall platform of the Alibaba Group. A summary of product reviews is given at the top of the page, with red indicating positive comments and green indicating negative comments. The summary is in the form of a set of key words and arranged, in descending order, based on the number of customers with similar opinions. When presenting this kind of summary, it is useful for customers to know the pros and cons of the product. This can greatly facilitate the consumer decision-making process.

Fig. 1. The screenshot of product review for Huawei Mate 9 in T-mall of Alibaba.

Keyword extraction is quite similar to the text summarization task in natural language processing. Text summarization has long been studied in the area of natural language processing [2]. It can be divided into two

elementary types: extractive summarization and abstractive summarization. Extractive summarization is the process of selecting informative or salient units from the source text data and concatenating them in an abridged version. In contrast, abstractive summarization involves generating novel sentences to represent the main content of the source from a more high-level perspective [13]. The original text is often transformed into abstract-like summaries by compressing, ordering, and merging selected units to maximize coherence and remove redundancy. Most proposed text summarization systems produce extractive summaries and transform long texts into several succinct sentences. Hori and Furui [3] proposed a method that calculates the maximum summarization score of a set of words, according to a target summarization ratio. The summarization score consists of a word significance measure and a linguistic likelihood, both of which are text-based features and extracted from transcripts. Kolluru et al. [4] proposed a series of multi-layer perceptions to summarize newscasts based on term-weighting and named-entity features. They found that their summarizer performed very well according to a question-answering evaluation and ROUGE analysis, but slightly less well on subjective fluency criteria. Zhu and Penn [5] used the maximal marginal relevance (MMR) score as a single feature in their summarization model, scoring a candidate sentence according to how generally relevant it is for a generic summary and how similar it is to the sentences that have been selected. Chen et al. [6] proposed the use of probabilistic latent topical information for extractive summarization. They verified the summarization capabilities by comparison with the conventional vector space model and latent semantic indexing model, as well as the hidden Markov model (HMM). Their experiments were performed on Chinese broadcast news, with noticeable performance gains obtained.

However, the keyword extraction task discussed in this paper is similar to the extractive summarization used in natural language processing, with important differences. The abovementioned methods output succinct sentences as the representation of the original text. However, our task only needs several keywords as the output. In a typical product review text chunk, there are only several relevant words. Our purpose is to identify keywords automatically and place them in the corresponding product feature category. For example, the following is a product review on a Braun razor on Amazon.com. In this long text chunk, only three are keywords relevant to the product's feature: "lighter," "comfortable to hold and use," and "outstanding shave." Clearly, traditional extractive summarization cannot handle this task.

'This Braun Series 7 7865 cc Wet and Dry electric shaver is the second Braun shaver I've purchased from Amazon.com. My prior Braun shaver is Series 7 790cc. I loved that one, but it was time for me to get a new one. I've used my new Braun shaver only one, but the shaver seems lighter and more comfortable to hold and use. Further, the shave is outstanding. I've used it only in dry mode. I highly recommend this shaver, and I'd buy it again. Hopefully, 5-6 years, or more, from now, Braun will have made further improvements.'

In addition, challenges persist in the product review summarization function, as shown in Figure 1. The summarized keywords are mostly pre-determined by the website. Each time a new user wants to write a review, he or she needs to tick the corresponding keywords as well. This process can be biased because the keywords are not extracted automatically based on the raw data and potentially important keywords may be ignored or missing. Indeed, keyword extraction is still at an infant stage for e-commerce, and is only popular on Chinese e-commerce sites such as Tmall, JD, and Taobao. Thus, it is necessary to develop an automatic and unbiased keywords extraction methodology from raw product review text.

This paper is prepared to overcome the aforementioned challenges by extracting the keywords from product reviews text. Currently, deep learning techniques have been proven successful in the areas of computer vision, natural language processing, etc, but have seldom been applied to the task of keyword extraction. We apply bi-directional LSTM RNN to process the product review text and extract keywords. Experiments on mobile review data are used to test our method's effectiveness, and significant results are obtained. The remainder of this paper is organized as follows. We introduce our research framework in section 2, and provide and discuss our Experimental results in section 3. Section 4 concludes the paper.

II. METHODOLOGY

Deep learning has gained much attention in recent years, as it yields state-of-the-art performance in many tasks [7]. Among the techniques available in deep learning, the long short-term memory (LSTM) recurrent neural network (RNN) is considered extremely suitable for sequential data such as speech, text and video [8]. As customer product reviews are in the form of text chunks, we use LSTM RNN for the keyword extraction task.

A. LSTM Network

The basic elements in an LSTM memory unit are three essential gates and a cell, as illustrated in Figure 2. All of the information memorized at time t is scored in the memory cell. The state of the memory cell is bonded with input three gates: the input gate, the output gate and the forget gate. Each gate's input consists of input and a recurrent part. The input gate is used to determine what new information should be stored in the LSTM cell. The output gate determines the content of output. The recurrent part is updated by the current status and feeds the information into the next iteration. The forget gate

decides which information from the previous iteration should be abandoned. [9].

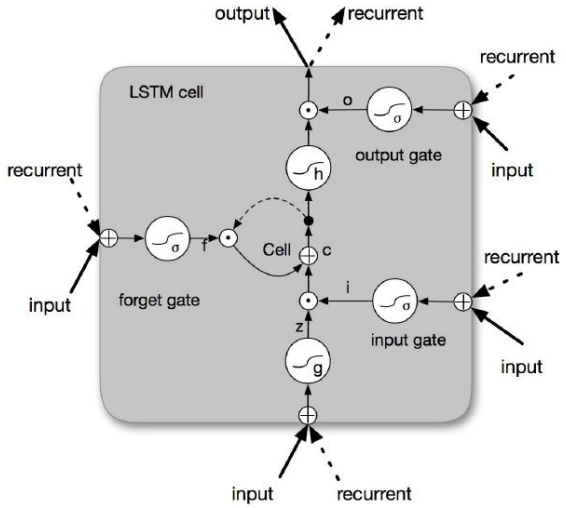

Fig. 2. LSTM unit [9].

Given a sequence of input vector $x = (x_1, ...x_T)$, the RNN computes the output vector $y = (y_1, ...y_T)$ as follows;

$$z_t = g(W_z x_t + R_z y_{t-1} + b_z)$$
$$i_t = \sigma(W_i x_t + R_i y_{t-1} + p_i \otimes c_{t-1} + b_t)$$
$$f_t = \sigma(W_f x_t + R_f y_{t-1} + p_f \otimes c_{t-1} + b_f)$$
$$c_t = i_t \otimes z_t + f_t \otimes c_{t-1}$$
$$o_t = \sigma(W_o x_t + R_o y_{t-1} + p_o \otimes c_t + b_0)$$
$$y_t = o_t \otimes h(c_t)$$

where $W_z, W_i, W_f, W_o, R_z, R_i, R_f, R_o$ are weight matrices for each gate's input and recurrent parts; b_z, b_i, b_f, b_o are the bias vector; σ, g, h represent nonlinear functions; $\otimes$ is point-wise multiplication calculation of two vectors; and p_z, p_i, p_f, p_o are used to denote the peephone connection in an LSTM network.

B. Bi-directional LSTM Network

Compared to an LSTM network, a bi-directional LSTM (BLSTM) network contains two parallel layers that propagate both forward and backward, thus allowing it to obtain information on the sequential series from both the past and future. Each forward or backward layer functions in a similar way to a regular LSTM, but the merit of the BLSTM is that it can incorporate the information of the series from both directions [10]. The output of BLSTM y_t is thus the concatenation of these two layers' output h_{ft}, h_{bt}, i.e., $y_t = [h_{ft}, h_{bt}]$ where

$$h_{ft} = H(W_{xh_f} x_t + W_{h_f h_f} h_{f_{t-1}} + b_{h_f})$$
$$h_{bt} = H(W_{xhb} x_t + W_{h_b h_b} h_{b_{t-1}} + b_{h_b})$$

C. Research Framework

1) *Filtering of product review raw data:* As in the Amazon review example given earlier, many sentences in review text chunks are irrelevant to the products being reviewed. Before the keyword extraction process, pre-processing is needed to filter out irrelevant sentences. This filtering process is considered a classification task. We manually annotate whether the sentence in the test chunk is relevant. Based on the annotated sentence, we build an LSTM RNN based classifier, which we use to determine the relevance of new sentences in the review text received from new customers. If a sentence is relevant, it will pass to the keyword extraction stage for further processing.

2) *Keywords extraction:* After filtering out the irrelevant product review information, the bi-gram, tri-gram and four-gram of the Chinese characters in relevant sentences are identified. Each n-gram appears with a different frequency. We sort all of the n-grams according to their frequency and use the top ones as the keywords of the product review.

The keyword extraction is a classification task in which the keywords are used as class labels. An LSTM RNN based classifier is trained to map the original sentences into the labels. In this way, we can obtain the frequency at which each label appears in the whole text chunk. Labels with high frequency are then used as keywords.

In summary, the research framework can be illustrated as Fig. 3.

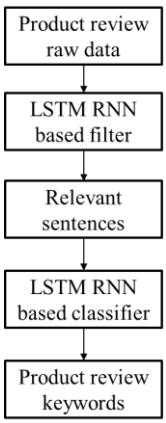

Fig. 3. Working flow of LSTM RNN based keywords extraction procedure.

III. EXPERIEMENTS

A. Experiment Data

Product review data were obtained by data-crawling jd.com, a major e-commerce platform in China. The data contain 15,866 reviews of six popular mobile phones, with 97,394 sentences in total[1]. The sentences serve as the text chunks in our task. After human annotation, 64,680 sentences are product relevant sentences, and the remaining 32,714 are irrelevant sentences. The sentences are used to train the LSTM RNN model to filter out irrelevant sentences. The tool of jieba (https://www.oschina.net/p/jieba) is used to segment the text at the word level.

B. *Experiment Results*

1) *Filtering accuracy*

The review data are filtered first so only relevant reviews are retained. As mentioned in section II, the LSTM RNN based classifier is used to achieve this goal. 10-fold cross-validation is used for model training and testing. If only the first 10 words are used in the training task[2], the average classification accuracy is 98.74%. We divide the whole data set into 10 subsets, each time we use seven of them to train the model and three of them as the test data. After numerating all possible combinations of training and testing data, the average classification is 98.4%, almost the same as 10-fold cross-validation.

The abovementioned results are based on Chinese word level classification. It is also of interest to test the methods based on Chinese characters. In this sense, word segmentation is not needed as with non-character languages such English. In addition, we use word2vec, a word embedding technique, to encode the words and characters to word vectors [11]. The data set is again divided into 10 subsets, among which seven are used for training and three for testing. The first 20 words (or elements in word vectors) are used as the training and testing features. The results are shown in Table I.

TABLE I
FILTERING ACCURACY BASED ON DIFFERENT FEATURES

Features	Classification accuracy
Character	99.5%
Word	98.9%
Word vector based on characters	66.4%
Word vector based on words	60.6%

The classification accuracy when using Chinese characters and Chinese words is comparable. This means that our filtering method can be applied to other languages, such as English, with a high level of accuracy, as there is no word segmentation process for natural language processing in English. Because the task is relatively straightforward and the raw data are for one particular product, i.e. smart phones, the results are good at the word and character level. Transforming the raw data into word vector does not help in this case. We anticipated that if more product review data are included, the word vector based filtering will catch up quickly.

2) *Keywords extraction accuracy*

As mentioned in Section II, the keyword extraction is considered a classification task. We use the top-seven frequent n-grams as the label of the mobile phone's features: appearance, cost-performance, multiple functions, smooth running, screen resolution, stand-by time and screen size. Then the LSTM RNN based classifier is trained based on the text chunk and the corresponding label. At this stage, the labels are assigned based on the sentence's meaning. For example, the sentences of "the mobile phone is smooth when running" and "the speed of the mobile phone is quick" both have the label "running smoothly." After pruning the sentences, 64,680 have the top-seven most frequent labels. These are used for model training and testing. Here, a 10-fold cross-validation is applied. The experiments are conducted at the character level and the word level, with word vectors based on characters and word vectors based on words. The results are shown in Table II.

TABLE II
KEYWORDS EXTRATION ACCURACY BASED ON DIFFERENT FEATURES

Features	Classification accuracy
Character	93.7%
Word	91.5%
Word vector based on characters	88.2%
Word vector based on words	89.5%

We can see that given a product review sentence, the proposed method has a high level of accuracy for assigning a class label (i.e., the keyword) to it. Once the classification model is obtained, we can use it to test the novel product review sentence. Each keyword will have a frequency of appearance in the whole testing set. The keywords can be sorted based on descending order of frequency and can be displayed to new customers to help them make better purchasing decisions. It should be noted that this classification task is relatively complicated compared to the filtering task, as it is based on the meaning of the sentence rather than on word similarity. Thus, the performance based on word vectors is much better than the filtering task. As a comparison, the performance based on the original characters and words is worse.

IV. CONCLUSION

Product review data provide valuable information for customers to better understand the product and make purchasing decision. However, information overload issue persists in product review data. This paper presents a LSTM RNN based approach to extract keywords from product review text. The results make customers quickly get the review opinion about the product and could greatly facilitate consumer decision making process. Provided

[1] Data are available upon request.

[2] If a sentence contains fewer than 10 words, all of them are used to train the model.

with the set of keywords generated from product review texts, relevant research can be conducted. For example, in the area of product customization, it has been acknowledged that there is a semantic gap between customer needs and product specifications, as customers may not have the expertise to identify each product specification accurately [12]. They may only express their needs in vague everyday language. The keywords extracted using our method are mostly in plain language and are quite similar to customer needs. A mapping function can be learned to connect customer needs (keywords generated from product reviews) to product specifications. Thus, given information regarding a new customer's needs, a potentially satisfactory product can be identified based on the mapping function.

Notwithstanding the importance of its findings, this paper still has some limitations. The experiment is conducted at a small scale, with only six products considered. In addition, we do not consider the problem of fake product reviews, assuming that all of the review texts are reliable. We will, however, address these issues in future work.

ACKNOWLEDGMENT

This work is partially supported by Hong Kong Research Grant Council, Faculty Development Scheme (UGC/FDS14/E02/15).

The authors contribute equally to this work.

REFERENCES

[1] T. Liang, X. Li, C. Yang, and M. Wang, "What in Consumer Reviews Affects the Sales of Mobile Apps: A Multifacet Sentiment Analysis Approach", *International Journal of Electronic Commerce*, vol. 20, no. 2, pp. 236-260, 2015

[2] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th International Conference on World Wide Web. ACM, ACM WWW 2003*, Budapest, Hungary, pp. 519-528.

[3] C. Hori, and S. Furui, "Advances in automatic speech summarization," in *Proc. EUROSPEECH2001*, vol. 3, pp. 1771–1774, 2001.

[4] B. Kolluru, Y. Gotoh, and H. Christensen, "Multi-stage compaction approach to broadcast news summarisation," *in Proc. of Interspeech 2005*, Lisbon, Portugal, 2005.

[5] X. Zhu and G. Penn, "Comparing the roles of textual, acoustic and spoken language features on spontaneous-conversation summarization," *in Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pp. 197–200, 2006,

[6] B. Chen, Y. Yeh, Y. Huang, and Y. Chen, "Chinese Spoken Document Summarization Using Probabilistic Latent Topical Information," *in Proc. ICASSP'*06, 2006.

[7] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[8] K. Greff; R. K. Srivastava; J. Koutník; B. R. Steunebrink; J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, 2017, accepted

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

[10] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp.2673–2681, 1997.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781, 2013

[12] Y. Wang, and M. M. Tseng, "A Naïve Bayes approach to map customer requirements to product variants," *Journal of Intelligent Manufacturing*, vol. 26, no. 3, pp. 501-509, 2015.

[13] J. Zhang, "Extractive speech summarization using structural modeling," Ph.D. dissertation, The Hong Kong University of Science and Technology, Hong Kong, 2011.