IIIT-H at IJCNLP-2017 Task 3: A Bidirectional-LSTM Approach for Review Opinion Diversification

Pruthwik Mishra[1]
IIIT-Hyderabad
Hyderabad, India
500032

Prathyusha Danda[1]
IIIT-Hyderabad
Hyderabad, India
500032

Silpa Kanneganti[2]
i.am+ LLC.
Bangalore, India
560071

Soujanya Lanka[3]
i.am+ LLC.
37 Mapex building
Singapore - 577177

Abstract

The Review Opinion Diversification (Revopid-2017) shared task (Singh et al., 2017b) focuses on selecting top-k reviews from a set of reviews for a particular product based on a specific criteria. In this paper, we describe our approaches and results for modeling the ranking of reviews based on their usefulness score, this being the first of the three subtasks under this shared task. Instead of posing this as a regression problem, we modeled this as a classification task where we want to identify whether a review is useful or not. We employed a bi-directional LSTM to represent each review and is used with a softmax layer to predict the usefulness score. We chose the review with highest usefulness score, then find its cosine similarity score with rest of the reviews. This is done in order to ensure diversity in the selection of top-k reviews. On the top-5 list prediction, we finished 3^{rd} while in top-10 list one, we are placed 2^{nd} in the shared task. We have discussed the model and the results in detail in the paper.

1 Introduction

With the increase in usage of e-commerce websites like Amazon, the views of the consumers on products that they purchase, have become both massive and vital to the on-line purchasing community. The facility to express one's views on a purchased product, helps the community members gain perspective on the features as well as quality of the product. This helps them in their decision making of buying the product. Hence, product reviews have tremendous effect on the sales of a product. Due to all these factors, it is very important for the sellers to know which reviews are immediately visible to the buyers. A review should not be given importance just based on its recency. For someone, it might not be very informative compared to not so recent or old reviews. It is vital that the top k reviews that are displayed to the customer are as descriptive as possible.

In view of the above, the IJCNLP shared task: Review Opinion Diversification focuses on ranking the product reviews based on certain criteria. The criteria is unique for each of the three subtasks under this shared task. Ranking must be done based on usefulness score in the case of Subtask A, where as in Subtask B, the goal is to rank the top-k, so as to maximize representativeness of the ranked list. In both Subtask A and B there should also be less redundancy among the top ranked reviews. The goal of Subtask C is to get the top-k reviews which will cover majority of the popular perspectives that are in the data. Similar to Subtask A and B, Subtask C should also have less redundancy among the ranked reviews. In this paper we have aimed for the Subtask A- Usefulness Ranking. Usefulness score of a review is the ratio of number of people who have found the review useful to the total number of people who have assessed the review as useful or not.

We built a model for the Usefulness Ranking subtask using neural networks. We have posed the ranking task as a regression problem in the early stage and then used cosine similarity to achieve the goal. We used bi-directional Long Short-Term Memory units (bi-LSTM) to get a representation of the reviews. Using these vector representations we obtained the top-k most useful, less redundant reviews for each product.

The paper is organized as per the following - section 2 explains the related work and details about the corpus. Different approaches employed are explained in the subsequent sections. Results

Proceedings of the 8th International Joint Conference on Natural Language Processing, Shared Tasks, pages 53–58,
Taipei, Taiwan, November 27 – December 1, 2017.

and Error Analysis constitute sections 5 and 6 respectively. The evaluation metrics are detailed in section 7. We conclude our paper with the Conclusion & Future Work section.

2 Related Work

(Zhou and Xu) dealt with classification of Amazon Fine Food reviews, based on usefulness score of the review. The classification of a review is binary, it will be classified as either useful or not useful. In their training data they have tagged a review as useful if it has been voted by at least on user as well as more than 50% percent of the users find it helpful. They have employed both feed-forward neural networks (Bishop, 2006) and LSTMs (Hochreiter and Schmidhuber, 1997) for classifying the products. In feed-forward neural networks, they used GloVe (Pennington et al., 2014) as embedding for word vectors and in LSTM model, self-trained word vectors were used to represent the reviews. The best feed-forward model had F1 score of 0.78 where as the LSTM model had 0.86.

In (Hu et al., 2017) a multi-text summarization technique is proposed. The idea is to identify the top-k most informative sentences to use them to summarize the reviews. The training data used are hotel reviews obtained from online sites like TripAdvisor. The novelty of the approach is to consider critical factors like author reliability, review time, review usefulness along with conflicting opinions. Their research method starts by collecting hotel reviews and then proceeds to review preprocessing. Next, sentence importance and sentence similarity are calculated by taking author credibility and usefulness scores into consideration. The last task is the selection of the top-k sentences, which involves grouping the sentences into k clusters, which is done by using k-medoids (Ester et al., 1996) algorithm. Human evaluation was done and the results showed that the proposed approach provided more comprehensive hotel information.

3 Corpus Details

The corpus provided for the task was extracted and annotated from Amazon SNAP Review Dataset[1] (McAuley and Leskovec, 2013). Examples of some reviews in the training data are given below-

- ’reviewerID’: ’A30O9Z1A927GZK’, ’asin’: ’B00004TFT1’, ’helpful’: [0, 0], ’reviewText’: ’Good price. Nice to have one charging when the other one is being used. They were more expensive in the stores, if you can find it.’, ’overall’: 5.0, ’summary’: ’Power wheels 12 volt battery’, ’unixReviewTime’: 1405209600, ’reviewTime’: ’07 13, 2014’
  - ’reviewerID’ gives the ID of the reviewer
  - ’asin’ is ID of the product reviewed
  - ’name’ field gives the name of the reviewer
  - ’helpful’ is the usefulness rating of the review which is a list of two numbers 1^{st} being the number of people who found the product helpful and 2^{nd} denotes the total number of people who accessed the review
  - ’reviewText’ is the text of the review
  - ’overall’ is the overall rating of the product
  - ’summary’ is summary of the review
  - ’unixReviewTime’ is the time of the review
  - ’reviewTime’ is time of the review (raw)
- ’reviewerID’: ’A1D9U33OHQTO18’, ’asin’: ’B00000016W’, ’reviewerName’: ’Julie L. Friedman’, ’helpful’: [0, 6], ’reviewText’: ’This album is the Beach Boys at their best. The genius of Brian Wilson, the beautiful voice of Carl Wilson, Denis Wilson on the Drums, Al Jardine with rhythm guitar and Mike Love with the Lyrics. A true classic. By Gregg L. Friedman MD, Psychiatrist, Hallandale Beach, FL’, ’overall’: 5.0, ’summary’: ’Gregg L. Friedman MD, Psychiatrist, Hallandale Beach, FL’, ’unixReviewTime’: 1343865600, ’reviewTime’: ’08 2, 2012’ This review got a usefulness score of 0 because 6 persons who accessed the review did not find the review useful.

  The training corpus details are shown in table 3

- Product_Type → Type of the product
- MaxLen → Maximum Length of a review

[1] https://snap.stanford.edu/data/web-Amazon.html

Product_Type	MaxLen	Non-Useful	Total Reviews	Total Products
Automotive	2974	112035	172016	569
Beauty	4399	185061	316536	1000
Toys Games	5853	217766	314634	1000
Grocery	5532	183146	293629	800
Video Games	7785	169455	358235	1000
Baby Products	5217	236950	352231	1000
Office	5043	195627	327556	1000
Patio Lawn	3963	152329	263489	859
Health	5566	200859	357669	1000
Tools Home	5939	205331	320162	1000
Digital Music	6397	56502	145075	468
Pet Supplies	5263	271616	398658	1000

Table 1: Training Corpus Details

- Non-Useful → the total number of reviews with a usefulness score $= [0, 0]$
- Total Reviews → total number of reviews of a product-type
- Total Products → Total number of products under a product-type

We did not include the Non-useful reviews defined above in training our network.

4 Approach

The main task here is to predict the usefulness of a review. The usefulness score describes the fraction of people who found the review useful. e.g. if 5 users have seen a review and 3 of them have found it useful, then the usefulness score $= 3/5 = 0.6$. The usefulness score is a continuous value.

We implemented a bi-directional LSTM (bi-LSTM) (Graves and Schmidhuber, 2005) for this task. The architecture and model details are explained in the following sections.

4.1 Architecture

Each review is modeled as a sequence of Glove vectors (Pennington et al., 2014). Bi-LSTMs present better semantic representations for a sequence where future and past information are encoded. Figure 4.1 shows the architecture of the bi-LSTM. We used glove embeddings trained on amazon reviews data for each word.

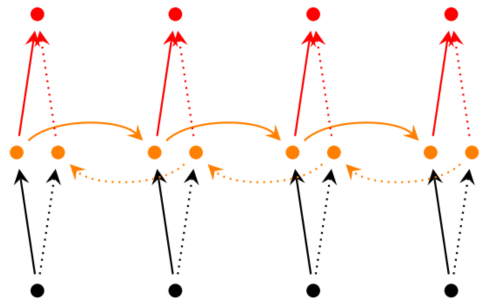

Figure 1: Bidirectional-LSTM [2]

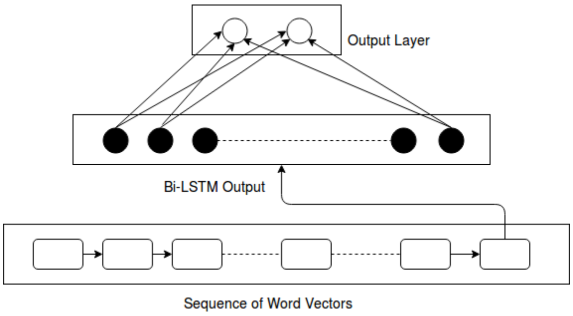

Figure 2: System Architecture

4.2 Model

We implemented a bi-LSTM using keras deep learning library (Chollet et al., 2015). We label any review as useful if its usefulness score is greater than 0.5, non-useful otherwise. The loss function used is binary cross entropy which is the most common loss function for binary classification tasks. Each glove vector is of 300 dimensions. The maximum sequence length is found from the the training set. The output layer uses softmax (Bishop, 2006) activation function. The dropout (Srivastava et al., 2014) rate in the network is 20% or 0.2. Adam (Kingma and Ba, 2014) optimizer was used to model the network.

[1] http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

System	mth	cos_d	cos	cpr	a-dcg	wt	unwt	recall
S1-Top-5	0.71	0.81	0.82	0.63	6.77	670.16	21.27	0.70
S2-Top-5	0.69	0.81	0.83	**0.73**	**6.88**	**681.75**	**21.54**	**0.73**
S3-Top-5	**0.79**	**0.82**	**0.84**	**0.74**	6.65	678.66	21.16	0.68
S1-Top-10	0.80	**0.89**	**0.91**	0.73	8.25	1768.09	54.77	**0.88**
S2-Top-10	0.77	0.88	0.90	**0.78**	**8.28**	1759.26	**55.14**	**0.88**
S3-Top-10	**0.84**	0.88	0.89	0.77	8.24	**1772.34**	54.99	0.86

Table 2: Results on Development Data

4.3 Scoring Measures

The main task here is to pick top-k reviews maximizing the usefulness score and minimizing the overall redundancy among the selected reviews. So we used 3 different weighting schemes where the hyper-parameters are tuned using Grid-Search technique. We used a linear interpolated weights for the overall usefulness estimate.

4.3.1 Basic Scoring Measure

In this scoring scheme, we sorted the reviews of any product based on it predicted usefulness score. The proposed model assigned a classification probability to any incoming review based on its usefulness. The probability score of deciding whether a review is useful or not is directly proportional to the usefulness of the review. The assignment of higher usefulness score to a review with high classification probability of being useful seemed a fair assumption. We chose top-N useful reviews (N=20 for this experiment). We pick the bi-LSTM representation of the review which has the maximum score. Then the cosine similarity between each of the rest of the vectors of reviews with the most useful review was evaluated. We picked the k-reviews which had least cosine similarity with the highest one. This ensured diversity as well as usefulness in the selected reviews. The cosine similarity between two sequences with Bag-Of-Words (BOW) representation many times fails to capture the semantic similarity between them. So the bidirectional-LSTM representation of a sequence which captures long range dependencies in the both the directions proved to be a better alternative.

4.3.2 Weighted Scoring Measure

To have a better weighted score with usefulness and diversity, we used two scoring measures which are explained below.

$$u_i = \alpha * p_i - \beta * cosim(v_{max}, v_i) \qquad (1)$$

- u_i → usefulness score of the i^{th} review
- p_i → predicted usefulness score of the i^{th} review
- $cosim(a, b)$ → Cosine-Similarity between vectors a and b, v_max →bi-LSTM output vector of the review with maximum usefulness score
- v_i →bi-LSTM output vector of i^{th} review
- α, β → tunable hyper-parameters

$$u_i = \alpha * (p_i) - \beta * (cosim(v_{max}, v_i) - 1.0) \quad (2)$$

The variables in equation 2 are the same as those defined in equation 1. We introduced an additional discounting factor while considering the cosine similarity between the vectors. The best values of the hyper-parameters were empirically found at $\alpha = 0.8$ and $\beta = 0.2$ through grid search cross-validation. Equation 2 refers to the relative difference between the cosine similarity between the most useful reviews and i^{th} review and the maximum possible cosine similarity score i.e. 1.0. The cosine similarity has nothing to do with the cosine similarity between overall vector representation of the review and opinion vector according to opinion matrix terminology defined by the evaluation system provided by the organizers.

5 Results

The results on the test and development data is tabulated in Table 2.

The systems described in tables 2 and 3 are defined as per the following.

- S1 → Predictions using basic scoring measure
- S2 → Predictions using scoring scheme defined in equation 1
- S3 → Predictions using scoring scheme defined in equation 2
- Top-5 → List of Top 5 predictions
- Top-10 → List of Top 10 predictions

We observed that the weighted scoring measures have a positive impact on most of the metrics used

System	**mth**	**cos_d**	**cos**	**cpr**	**a-dcg**	**wt**	**unwt**	**recall**
S1-Top-5	0.78	**0.86**	**0.87**	0.49	4.27	494.03	14.04	**0.76**
S2-Top-5	0.78	0.85	0.86	**0.52**	**4.34**	**495.35**	**14.34**	0.75
S3-Top-5	0.78	0.84	0.85	0.51	4.11	486.51	13.35	0.72
S1-Top-10	0.81	**0.92**	0.93	0.61	5.18	**1325.2**	37.54	0.89
S2-Top-10	**0.84**	0.91	0.92	**0.65**	**5.2**	1318.8	**37.8**	0.9
S3-Top-10	0.83	**0.92**	**0.94**	**0.65**	5.16	1317.5	36.8	**0.92**

Table 3: Results on Test Data

for evaluation. The evaluation of the mentioned metrics are done against an opinion matrix for each product. This opinion matrix has been created by evaluators. The evaluation code was provided by the shared task organizers.

6 Error Analysis

There are some reviews where the 'reviewText' field is blank. The input sequence in this case were a series of zero vectors. The usefulness score for these reviews were wrongly predicted. There were cases where the system incorrectly assigned highest probability score to non-useful review.

- 'reviewerID': 'A3S3HYY3BDTYA7', 'asin': 'B00003XAKR', 'reviewerName': 'Shellzbell', 'helpful': [0, 0], 'reviewText': 'Love this thing.. what a bed saver... I was finishing potty training my 2 year old and bed time was my biggest concern. But with this I do not have to worry about the foam mattress I have on my daughters bed. It is easy to wash and put back on the bed.. love this thing.', 'overall': 5.0, 'summary': 'What a find', 'unixReviewTime': 1365292800, 'reviewTime': '04 7, 2013'.

- This review is given highest usefulness during the prediction. Then an incorrect list of reviews were returned which was not be representative of any product.

The usefulness score for a review is very subjective. e.g if 6 out of 7 people have found a review useful, then the usefulness score $= 6/7 = 0.86$. If 1 out of 1 person found one review helpful, it is assigned higher score $1/1 = 1.0$ compared to the previous review. So the usefulness score should not be concerned only with the usefulness score, but it should also take into account the total number of people who access a review.

7 Evaluation

The organizers provided an evaluation system (Singh et al., 2017a) for evaluating the performance of the submitted systems [3]. There were different evaluation metrics for different subtasks. Those evaluation metrics are briefly described here

- SubTaskA
  - mth (More Than Half's) - The fraction of reviews where more than half of the reviewers voted in favour of them.
- SubTaskB
  - Cosine Similarity - The cosine similarity between the overall vector and the opinion vector. The opinion vectors were designed by human evaluators. This vectors are different from the vector representation used after training our bi-LSTM network
  - Discounted Cosine Similarity - The cosine similarity between the overall vector and the discounted opinion vector.
  - Cumulative Proportionality - This metric is based on Saint Lague method and widely used in Electoral Seat Allocation (Dang and Croft, 2012)
  - Alpha-DCG - This measures the diversity and novelty in ranking (Clarke et al., 2008)
  - Weighted Relevance - This is a discounted cumulative gain where the relevance of a review is evaluated by summing the weights of the opinions expressed in the review
- SubTaskC
  - Unweighted Relevance - This metric captures a discounted sum of number of opinion covered in a ranked reviews list
  - Recall - This is a measure of how many opinions are actually covered out of all possible opinions in the ranking

[3] https://sites.google.com/itbhu.ac.in/revopid-2017/evaluation

8 Conclusion & Future Work

In this paper, we showed that bi-directional LSTMs perform decently for a task of ranking the top-k reviews based on their usefulness score. This showed that a sequence of word vectors presented a good alternative for training systems without any hand-crafted features.

We can remove the blank reviews and train our system for further analysis. We intend to use character embedding along with the word embeddings to get better representation of a sequence, in this case a review. This will also help in getting a better representation for out-of-vocabulary(OOV) words. We can also include some linguistic regularization (Qian et al., 2016) while learning the bi-LSTM to take advantage of intensifiers, negative words, positive words, sentiment words and other cue words.

Acknowledgements

We thank Vandan Mujadia and Pranav Dhakras for their valuable inputs and feedback for us on the paper.

References

Christopher M Bishop. 2006. *Pattern recognition and machine learning*. springer.

François Chollet et al. 2015. Keras. `https://github.com/fchollet/keras`.

Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM.

Van Dang and W Bruce Croft. 2012. Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 65–74. ACM.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ya-Han Hu, Yen-Liang Chen, and Hui-Ling Chou. 2017. Opinion mining from online hotel reviews–a text summarization approach. *Information Processing & Management*, 53(2):436–449.

Diederik Kingma and Jimmy Ba. 2014. Adam a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Qiao Qian, Minlie Huang, and Xiaoyan Zhu. 2016. Linguistically regularized lstms for sentiment classification. *arXiv preprint arXiv:1611.03949*.

Anil Kumar Singh, Avijit Thawani, Anubhav Gupta, and Rajesh Kumar Mundotiya. 2017a. Evaluating Opinion Summarization in Ranking. In *Proceeding of the 13th Asia Information Retrieval Societies Conference (AIRS 2017)*, Jeju island, Korea.

Anil Kumar Singh, Avijit Thawani, Mayank Panchal, Anubhav Gupta, and Julian McAuley. 2017b. Overview of the IJCNLP-2017 Shared Task on Review Opinion Diversification (RevOpiD-2017). In *Proceedings of the IJCNLP-2017 Shared Tasks*, Taipei, Taiwan. AFNLP.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Zhenxiang Zhou and Lan Xu. Amazon food review classification using deep learning and recommender system.