Aspect Based Abstractive Review Summarization UsingBi-directional Gated Recurrent Unit

Khalid Mahmood[1] Fazal Masud Kundi[1] , Ghulam Muhammad Kundi[2] and Mahwish Kundi[3]

[1] Institute of Computing and Information Technology, Gomal University, Dera Ismail Khan, Pakistan,
[2] Faculty of Law and Administrative Studies, Gomal University, Dera Ismail Khan, Pakistan,
[3] Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan.

Summary
Text summarization produces a compressed version of the original document by selecting the most important contents. Text summarization is regarded as a generic technique because it did not provide any distribution of opinions and their expressed sentiments. Review summarization provides its user with the description of all product aspects or features with their sentiments or feelings, expressed in reviews which aid the online customer to judge the product or service and helps in decision making process. Review Summarization is considered as difficult task, due to unstructured behavior, review short length, and free style writing. This work proposes an aspect based abstractive summarization of customer reviews using encoder decoder architecture with attentions and pointer generator network. We have used Bi-directional Gated Recurrent Units (Bi-GRU) for encoder-decoder architecture to ensure that the adjoining words have influence on the resultant summaries. We have achieved ROUGE-I with 35.32 score, ROUGE-II with 38.46 score and ROUGE-L with 29.26 score on the Amazon reviews dataset. Empirical results indicate the efficacy and efficiency of our suggested model with respect to the base line systems.

Keywords:
Deep Learning , Automatic Summarization, Aspect based Abstractive Summarization, Attention Mechanism, Pointer Generator Network.

1. Introduction:

Automatic Summarization of digital contents is a process to pick or select the most essential concept and points from the original document. It is considered as a solution to digest and distill the enormous amount of information available through the internet forums by saving the precious time of online community.

Automatic summarization was initially proposed by Luhn in 1950’s [1] to get rid of manual summary writing approach. [2] suggested a relative-frequency method for measuring the significance of words in the text, phrases, and sentences to address the read and analysis of report by scanners. Due to unprecedented growth of social media contents on the internet, it is now impossible for the individual or an organization to extract the information from this sheer bulk of data. An approximate, “over 2.5 quintillion bytes of data is produced on daily basis” [3]. The automatic summarization is suitable for search engines to find the relevant information from the summaries rather than searching the whole document. The Automatic summaries are categorized on the basis of Output, which classifies the summaries into Extractive summaries and Abstractive summaries. On the basis on Input, the Automatic summary could be generated from single document or it can be generated from multiple documents. On the basis of content, the Automatic summary could be domain specific [4-6], Generic Summaries [6-9] and Query based summaries [10-14]. Two approaches towards automatic summarization are proposed by the research community: Extractive vs. Abstractive Summarization. The extractive summarization approach, extracts the important information from the document [15], whereas the abstractive summarization tries to mimic the human summary generation process [16]. Different algorithms are proposed and suggested by the research community on both versions, producing state of the art results[17].

Manuscript received June 6, 2020
Manuscript revised June 20, 2020

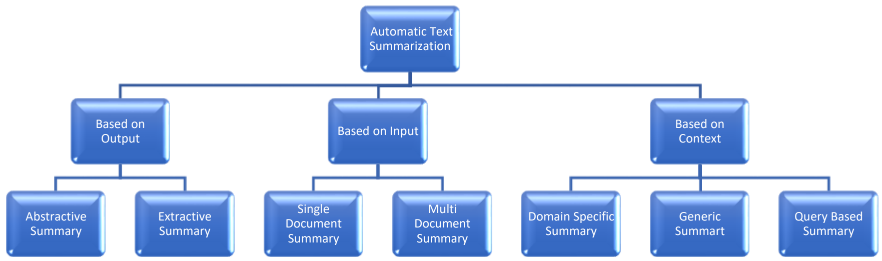

Fig. 1 Automatic Text Summarization taxonomy.

The Abstractive summarization approach is considered as a difficult approach with respect to its counter part due to the generation of summary, as produced by the humans, as it did not cut and glue the summary contents, done by the extractive summarization approach, but considers the deep and semantic understanding of the linguistic phonemes, and tries to rephrase and generate those contents, which are not present in the actual document.

Neural Network models are “Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains” [18]. The development of Feed Forward Neural Networks (FFNN) was started in the late 1950’s by the [19] perceptron idea. The drawback of FFNN was the flow of information in only forward direction. The Recurrent Neural Networks (RNN’s) were developed by [20] which are based on Hopfield network, a special kind of RNN discovered by the [21]. The RNN’s network allows long term dependencies in which the output at any time, not only depends on the current input but also on the output produced at previous step. RNN’s contains loops, which allow the information to be stored within the network. In short, the RNN’s use their reasoning from previous experience to inform the upcoming events. The vanishing /exploding gradient problem of vanilla RNN’s for the long sequences was addressed in [22] by proposing Long Short term memory (LSTM) and Gated Recurrent Units (GRU) architecture proposed by [23] . Both structures have gates, which decides to either forget or to keep information for future context. The GRU has similar functionality compared to LSTM but has fewer parameters and gates than LSTM.

Sequence to sequence models using Deep learning algorithms have shown promising results in different Artificial Intelligence (AI) tasks like Machine translation [24], Speech-recognition [25], Game Programming [26], Vehicle Make, color and model Identification [27], Face Recognition [28], Image captioning [29], and Abstractive Text Summarization [30] to name a few. Sequence to sequence (S2S) model was proposed by [31] which accepts word sequence as input and produces word sequence as output. The success of “encoder-decoder framework in statistical machine translation”, has motivated the research community to apply neural language model to the abstractive summarization domain. Encoder-decoder (S2S) model were applied to abstractive summarization problem by [16, 32-36]. [32] have implemented attention-based encoder-decoder model to abstractive text summarization and yields remarkable performance to sentence level summarization dataset. Similarly [16] introduce “off-the-shelf attention encoder-decoder RNN” that apprehended hierarchy in the document structure and recognized the most important keywords and sentences within the document. [33] suggested the solution to “Out of Vocabulary (OOV)” words present in the output summary by proposing a pointer-generator network that can be extractive as well as abstractive in a sense that it can copy the words from source document and generate words from the fixed vocabulary. [35] in their work suggested a review summarization model based on the aspects and sentiments, but used Uni-direction LSTM, which only cover context in one direction and did not cover past as well as future context for encoded input for summary generation process.

1.1 Encoder-Decoder Architecture:

The Encoder-Decoder Architecture using S2S model was first proposed for machine translation [37], it was then proposed to summary generation by [32]. The Encoder part takes the current input with previous hidden states to construct a context vector of all-time stamps. The decoder takes the context vector as initial input to generate the output summary. The difference between the expected output and ground truth is called as Loss function. The output from the decoder is propagated back to encoder to adjust weights during training phase to minimize loss functions.

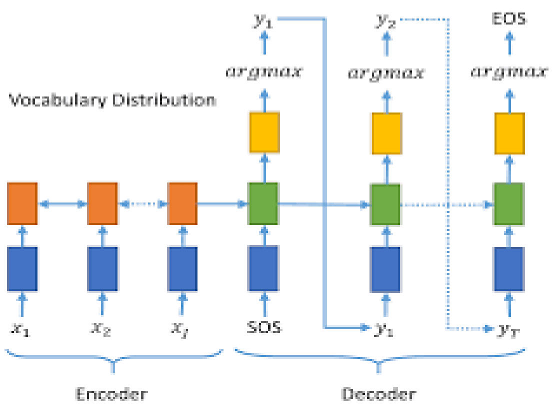

Fig. 2 Encoder-Decoder Architecture

We contribute in the following ways:

(i) We have proposed an algorithm for Aspect-based Abstractive Review Summarization (AARS).

(ii) We have used Encoder Decoder Architecture with Bi-GRU's for AARS.

(iii) We employ Teacher forcing technique, for optimization task, to moderate the exposure bias issue.

2. Literature Review:

[38] were the first one to propose an abstractive opinion summary for customer reviews. The authors use a Graph data structure to produce review summaries of abstractive nature for highly replicated text(s). The developed model is named as 'Opinosis'. Their proposed method generates ultra-concise summaries of hotels and cars reviews, which are tailored for small screens like PDA's and Cellular phones.

The work of [39] is a mixed mode sentiment summarizer which combines both the extractive and abstractive summarization techniques. This study proposes three summarizer version: Starlet-E (Extractive), Starlet-A (Abstractive) and Starlet-H (Hybrid). The summarizer selects "quotes from the input reviews" using extractive summarizer and then implants them into an abstractive summary to estimate the polarities as positive or negative for opinions. The limitation of this work is a lot of training data is needed to learn and further it considers a small number of aspects as inputs.

In this paper the authors [30] proposed an Abstractive summary generation framework. Initially Discourse trees (DT's) are generated from input reviews by applying Discourse relations using Rhetorical relations. Rhetorical relations are based on Rhetorical Structure Theory (RST). After that Aspect trees are generated from discourse tree, which contain aspects as leaf nodes, and then Aspect hierarchical trees are formed by applying weighted Page-rank algorithm to retain most important aspects. Finally, the Abstractive summaries are generated by applying Natural language generation techniques. Microplanning and sentence realization steps are performed at this phase. While quantitative evaluation with MEAD* and MEAD Lex-Rank describes the better efficiency of the proposed approach.

Neural approaches using Encoder-Decoder Architecture for summary generation were first addressed in [32]. In this research paper the authors suggested an abstractive sentence summary generation by applying neural encoder-decoder framework. In this work the input sentences are encoded by three embedding methods by BOW representation, convolutional encoder, and attention-based encoding. The output summary is generated by neural decoder using beam-search algorithm. The results are compared with benchmark studies and shows improvement in ROUGE metric scores.

Summary generation process is casted as Sequence to Sequence learning process in this research study [40]. The Gated Recurrent Networks (GRU) concept overcomes the drawback of simple RNNs. The vanishing or exploding gradient problem is resolved by using GRUs.

[16] proposed attention mechanism, which direct the decoder to pay attention to encoder part for summary generation using Attentional RNN encoder-decoder network. The attention mechanism proposed by [41] elevates the problem of predicting the output from a fixed window consisting of previous inputs

The issue of Unknown "UNK" tags: those words which are not present in the vocabulary were addressed in [33] by proposing abstractive text summarization through sequence-to-sequence model using pointer generator networks. In this paper the author addresses the two issues related to textual summary generation, the first one is the inability to produce factual information correctly and the second issue is replication of produced contents. The solution to the above cited problems is proposed through a pointer mechanism, which can produce new words with the restriction to not produce out of vocabulary (OOV) words, and moreover the coverage problem was handled by restricting the output for repeated words or attended words.

In this paper the author [35] proposed a sentiment aware abstractive review summarization (MARS) using LSTM encoder-decoder neural network with multi-factor mutual attentions. The input reviews are embedded as vector form for LSTM encoder with attention which learns the representations of context words, sentiment words and aspect words , and then abstractive summary is generated using attention fusion network thorough LSTM decoder. The empirical results show that the generated summary outperforms with its competitors. The limitation cited in this work is to use Uni-direction LSTM for summary generation.

The authors [36] in their word proposed an abstractive text summarization with an attention on reader aware comments to generate a summary that focus on the aspects/comments. In the first step the document along with user comments are given as input to encoder, which transforms it into embedded state/form. In the second step the decoder generates the summary with an attention on the user focused comments.

In this paper the authors [42] proposed abstractive template-based summary generation framework by applying three content selection and structuring methods: Rhetorical Structure Theory(RST), Concept Net and Hybrid technique by combining the above stated two methods. The limitation encountered is template design for each domain.

In this paper the authors [43] presented a data-to-text generation systems, in which an abstractive textual summary is generated from input data presented in tabular form.

3. Methodology:

Even though the extraordinary evolution of previous research studies in the realm of abstractive text summarization, generating aspect and sentiment aware summaries of product reviews continues to be an open contest in the real-world for two reasons. (i) First, the neural Seq2Seq model has a tendency to generate a standard summary, which contains high frequency phrases but did not contains the aspects or features with sentiment information from the reviews, which perform a critical role on the customer decision making process. Secondly the output produced by the abstractive summarization process have low ROUGE scores. Finally, the summary style and words in different categories and topics can significantly vary. However, the existing approaches applies a standardized model to generate text summaries, often missing the main aspects discussed in reviews. To improve the aforesaid issues and limitations, we design an abstractive review summarization model based on the aspects with sentiments . The workflow of the proposed methodology is depicted in the following diagram.

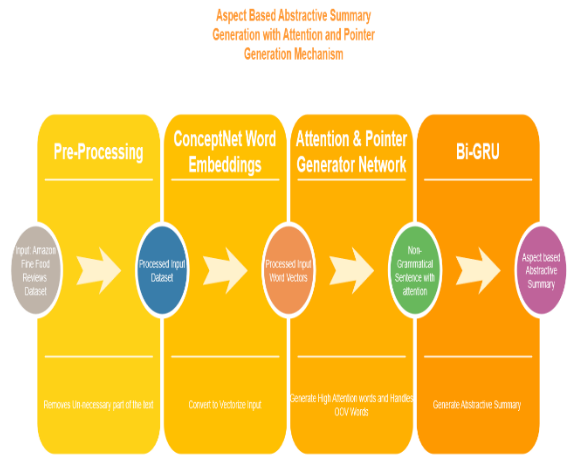

Fig. 3 Proposed Work-Flow diagram for Aspect-based abstractive review summarization.

The aspect-based abstractive review summarization using Seq2Seq model using Bi-LSTM is divided into four major modules. The first module performs the Pre-processing, which filters the input reviews document to replace contractions with their regular form, and after that special characters like “@, #, ?, /” are remove from the reviews text, stopwords removal are applied to the review text, as stopwords did not play any role towards further text processing tasks. The second module converts the review document along with their extracted aspects and opinion pairs with their sentiments into lower order word embeddings, to prepare it for the encoder input. We have used ConceptNet Numberbatch 3.0. word embeddings [44] , as it is considered as better one with respect to Glove or word2vec. In the third module the encoder outputs (Concept Vector) are given to decoder to generate summary. The decoder takes the concept vector as its initial input to generate summary. Our decoder module uses attention mechanism and pointer generator network to avoid the problems of encoder-decoder architecture. The attention mechanism addresses the problem of fix length concept vector of encoder part by paying attention to all parts of the input, while generating summary. The Pointer generator network resolves the issue of “out of vocabulary (OOV) words”, by placing a switch on top of the decoding step, to either generate the new word from vocabulary or copy the word from the original document. In the last module an aspect-based abstractive summary is generated, which highlights the major aspects and opinions of the customers.

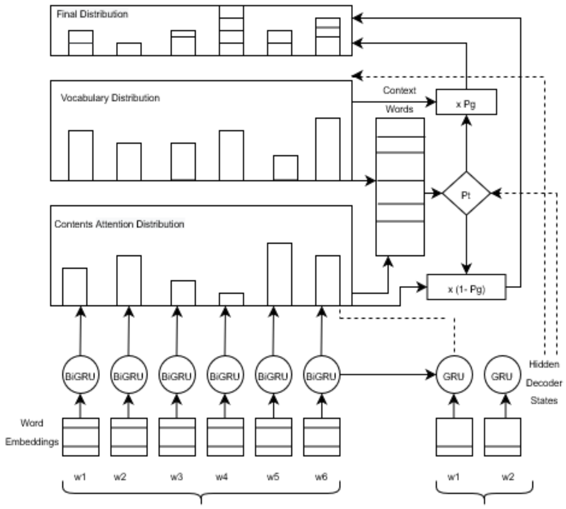

Fig. 4 Aspect-based Abstractive Review Summarization

3.1 Dataset Acquisition:

We have selected Amazon Fine Food Review Dataset, hosted on Kaggle [45]. This dataset contains nearly 500,000 reviews collected from amazon.com on a data span of 10 years, up to October 2012. These reviews include the complete information which consists of reviewer information, product description, customer review and its summary with time information.

3.2 Pre-processing:

The preliminary step is Pre-processing to prepare and arrange data ready for further processing by deep learning model. After reading the dataset file, the un-necessary data fields, like product information and description, reviewer information and review time are dropped from the reviews, and we only keep the “Review Text and Review Summary” information. The contracted words are substituted with their normal form in the Contraction removal step. The review text and summary are then cleaned to remove noise from the data and finally stopwords are removed from the contents, because they did not serve any purpose required by the natural language processing.

3.3 Word Embeddings:

The Natural Language Processing (NLP), a sub-field of Machine Learning (ML), deals with the natural language understanding and generation, which often consists of textual data, which in turn consists of language phonemes, consists of characters and words [46]. To process the textual data by any ML algorithm, the data must be converted into numerical form, because a machine cannot understand the textual data as human understand. The first representation, which is called one-hot vector, assigns a “1” for presence and “0” stands for the word absence in the vocabulary. One-hot vector representation suffers problems like large vocabulary space requirements and lacks for similarity issue or semantic meanings. Suppose for 10,000 words we need a matrix of 10,000 entries, Similarly Sparseness is another issue, in which a matrix contains “1” for only word presence and the remaining entries should contains zeros in all fields. The “Similarity issue”, in which to determine the similarity between two related cannot be determined with one hot vector. So, to handle the issues of one-hot representation , word embedding is used [47]. ConceptNet Numberbatch 3.0 is used for word embeddings.

3.4 Bi-GRU Encoder Layer:

The Bi-GRU layer is an advance version under the category of RNN, which is adept to holds long term dependencies [48]. It helps in access to both the prior (left) and later (right) context of an encoded reviews. The limitation to consider only previous context, without considering the next context, of hidden states in Uni-directional LSTM [35] has been overcome by the Bi-GRU, in which not only the previous context but forward (next) context to calculate the hidden states is considered. The hidden state at any time stamp can be calculated by the present input and the information from prior hidden states.

$$h_t = GRU(x_t, h_{t-1}) \quad (1)$$

In Eq.1 h_t is the encoder hidden state. x_t represents word embeddings at time stamp t and h_{t-1} represents the previous hidden state.
The Encoder state GRU formulas are shown in detail as under:

$$h'_t = tanh(W_c[z_r * h_{t-1}, x_t] + b \quad (2)$$
$$Zu = \delta(W_u[h_{t-1}, x_t] + b) \quad (3)$$
$$Zr = \delta(W_r[h_{t-1}, x_t] + b) \quad (4)$$
$$h_t = Zu * h_{t-1} + (1 - Zu) * h'_t \quad (5)$$

Where memory unit value is represented by h'_t, and Update and Reset gates are represented by Zu and Zr, respectively [23].
The Bi-GRU is combination of two GRUs calculated in forwards and in reverse direction and can be calculated as a concatenation of both directions. Bi-GRU can be formulated as:

$$\overleftrightarrow{hi} = \overrightarrow{ht} \oplus \overleftarrow{ht} \quad (6)$$

All encoder states are grouped into a single context vector, which are input to the decoder for summary generation process.

3.5 Decoder Layer:

The Decoder layer are designed to predict one word at each time stamp. Uni-directional GRU is proposed for the decoder layer. The Attention mechanism [41] restricts the decoder to pay attention to the encoder part while generating the summary, which results in generating those words which are most important for summary contents. The pointer generator network [33] generates the out of vocabulary word (OOV) issue. Teacher forcing mechanism is also included as policy gradient mechanism for loss function.

The formula to generate summary for decoder each state is:

$$s_t = GRU\ (y_{t-1}, c_t, s_t) \tag{7}$$

In eq (7): s_t shows the hidden state of the decoder, whereas c_t is the context vector and y_{t-1} is the decoder output and previous time stamp. The in-detail formula used for the decoder are as under:

$$h'_t = tanh(WEy_{t-1} + U[z_r * s_{t-1}] + Cc_t \tag{8}$$

$$z_u = \delta(W_u Ey_{t-1} + U_u s_{t-1} + C_u c_t) \tag{9}$$

$$z_r = \delta(W_r Ey_{t-1} + U_r s_{t-1} + C_r c_t) \tag{10}$$

$$s_t = Zu * s'_t + (1 - Zu) * s'_{t-1} \tag{11}$$

In the above equations, s'_t represent the memory unit value, z_u and z_r shows the update and reset gates [49].

The formula to calculate each contextual vector are

$$c_t = \sum_{i=1}^{M} a_{t_i} h_i \tag{12}$$

$$a_{t_i} = \frac{\exp(e_{ti})}{\sum_{i=1}^{T_x} \exp(e_{ti})} \tag{13}$$

$$e_{ti} = V_a^T \tanh(W_a s_{t-1} + U_a h_i\ ) \tag{14}$$

Where e_{ti} is an alignment model for i-th input at t-th position. a_{t_i} is the attention probability for i-th input at t-th position. The contextual vector is calculated as Summation of attention weights over encoder, multiplied with all hidden states. The context vector c_t is then concatenated with the decoder state s_t and fed through a linear layer and a SoftMax layer to compute the output probability distribution over a vocabulary of words at the current state.

$$Pvocab(w_t) = softmax(V[st, ct] + b) \tag{15}$$

Where number of rows in V represents the number of words in the vocabulary.

On top of the GRU decoder, we adopt the copy mechanism [33] to integrate the attention distribution into the final vocabulary distribution which is defined as the interpolation between two probability distribution.

$$P(w_t) = p_{gen} Pvocab(w_t) + (1 - gen) \sum_{i=1}^{M} a_{t_i} \tag{16}$$

Where $p_{gen} \in [0,1]$ is switch variable to control generating a word from the vocabulary or copy it from the source document. If w is an OOV word, then $Pvocab(w)$ is zero, $\sum_{i=1}^{M} a_{t_i}$ is zero if w does not appear in the source document. p_{gen} can be defined as:

$$p_{gen} = \delta(\ c_t + s_t + y_{t-1} + b) \tag{17}$$

3.5.1 Attention Mechanism:

Neural network tries to mimic the human brain actions in a simplified manner, Attention mechanism is also an attempt to implement the same action of selectively concentrating on a few relevant things, while ignoring others in deep neural network. In calculating the hidden states of an encoder input, all the hidden states are accumulated into a single context vector, but [41] proposed that while creating the context vector, emphasis should be put on embeddings of all the words, and it is simply done by taking a weighted sum of the hidden states. The context vector c_i for the output word y_i is generated using the weighted sum of the annotations.

3.5.2 Pointer Generator Network:

An inherent problem with abstractive text summarization approach is that the summarizer reproduces factual detail incorrectly. The Pointer generator network proposed by [33] can think as a hybrid approach by combing both extractive (pointing) and abstractive (generative) methods.

3.6 Teacher Forcing:

Teacher forcing is a technique which is applied to different neural network problems and yields better results. Teacher forcing technique is suitable for sequence to sequence problems, where without the teacher forcing technique, the wrong word prediction at first stage will propagate to next stages and will lead to wrong word predictions, which will definitely produce wrong results. While using teacher forcing technique the result of first generated word is compared with ground truth, and after checking its correctness, and if it is correct, it is passed to next stage for word generation. So, in teacher forcing, at each stage the output is compared with ground truth, and if it is wrong, then after its correction (teacher forcing), the output is propagated to next stage for word generation.

4. Experiments

4.1 Applying an Example

We take a sample review text from Amazon fine food review dataset to generate an abstractive aspect-based summary:

"I've bought several of the Vitality canned dog food products and have found them all to be of good quality. The

product looks more like a stew than a processed meat and it smells better. My Labrador is finicky, and she appreciates this product better than most",
and its summary: *"Good Quality Dog Food".*

4.2 Pre-processing

- Contraction Removal

"I have bought several of the Vitality canned dog food products and have found them all to be of good quality...... The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky, and she appreciates this product better than most."

- Data Cleaning

In this step we will remove un-wanted characters like "@, %, URL, ?" stop-words and deplete the text into lower form.
Clean Review # "1 bought several vitality canned dog food products found good quality product looks like stew processed meat smells better labrador finicky appreciates product better"
Clean Summary # 1 "good quality dog food"

4.3 Embedding Layer:

After normalizing the text, we segmented the input review and summary into set of tokens as follows by just showing only seven (7) words : "bought", "several", "vitality", "canned", "dog", "food", "products". A vocabulary index is built which maps the index to each distinct word. The vocabulary index assigns "bought: 1", "several: 2", "vitality: 3", "canned: 4", "dog: 5", "food: 6", "products: 7". After vocabulary built, the given review is converted into a sequence of indices: [1,2,3,4,5,6,7]. Embedding layer transforms each word in the review with a single index into a vector form. We have used ConceptNet NumberBatch embedding, represented as follows: bought [0.0092 0.1695 -0.0663 -0.1317] several [0.0246 0.102 -0.0625 -0.0157] vitality [0.0335 0.0652 -0.058 0.0305] canned [0.0676 0.0416 -0.0468 0.1589]. where the first rows signify an embedding representation for the token "bought", and the second row signifies the embedding representation for token "several". The process is repeated for each token for the review text and summary. Resultantly, a matrix is constituted as follows; [[0.0092 0.1695 -0.0663 -0.1317], [0.0246 0.102 -0.0625 -0.0157] , [0.0335 0.0652 -0.058 0.0305], [0.0676 0.0416 -0.0468 0.1589]. The ConceptNet Numberbatch embedding layer has 300 dimensions for each word.

4.4 GRU Encoder Layer:

The embedding layer output is provided as input to the Bi-GRU Encoder layer. The computation is comprised of four component parts, namely a reset *(z_r)*, update *(z_r)* gates, as well as new memory container *(c~t)*.

4.4.1 Forward GRU

In Forward GRU layer each gate takes input in the form of current input *(x_t)*, prior hidden states *(h_{t-1})*, performs some computation and finally information is aggregated in the form of hidden state "$\vec{h}$", as follows.

$$\begin{array}{ccccc} \vec{h} & & o_t & & c_t \\ \begin{pmatrix} 0.6 \\ 0.4 \\ 0.5 \\ 0.7 \\ 0.8 \end{pmatrix} & = & \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} & \odot\ \tau & \begin{pmatrix} 0.7 \\ 0.4 \\ 0.6 \end{pmatrix} \end{array}$$

4.4.2 Backward GRU

In Backward GRU Layer, each gate accepts a current input *(x_t)*, future hidden state *(h_{t+1})*, performs some computation and finally information is aggregated in the form of hidden state "$\overleftarrow{h}$", as follows:

$$\begin{array}{ccccc} \overleftarrow{h} & & o_t & & c_t \\ \begin{pmatrix} 0.3 \\ 0.2 \\ 0.2 \\ 0.1 \end{pmatrix} & = & \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} & \odot\ \tau & \begin{pmatrix} 0.4 \\ 0.3 \\ 0.2 \\ 0.1 \end{pmatrix} \end{array}$$

4.4.3 Final output of Bi-GRU Layer

Both the Forward GRU "$\vec{h}$" and Backward GRU "$\overleftarrow{h}$" are accumulated (element-wise summation) using Eq.6, to obtain the final representation $\overleftrightarrow{h}$, represented as follows:

$$\begin{array}{ccccc} \overleftrightarrow{h} & & \vec{h} & & \overleftarrow{h} \\ \begin{matrix} 0.9 \\ 0.6 \\ 0.7 \\ 0.8 \end{matrix} & = & \begin{pmatrix} 0.6 \\ 0.4 \\ 0.5 \\ 0.7 \end{pmatrix} & \oplus & \begin{pmatrix} 0.3 \\ 0.2 \\ 0.2 \\ 0.1 \end{pmatrix} \end{array}$$

4.5 Uni-Directional Decoding Layer

The context vector is inputted to the decoding layer to generate the aspect-based abstractive summary. At each decoding step, the previously decoded word is also supplied to the next decoding step along with new input to predict the probability to generate new word. The attention mechanism and pointer generator network is also included at the top of decoding layer.

4.6 Hyperparameter Settings

Iterations = 70000
hidden_size = 128
batch_size = 64
teacher-forcing-ratio = 0.5
num_layers = 2
learning_rate = 0.03
keep_probability = 0.95

4.7 Output Summary:

After training the model for 70000 iterations, and when the model stop training, the following review is input to the system, and the system generates the review summary.

Sentence: *once more amazon was great the product is good for kids even though it has a little bit more sugar than needed*
Predicted Summary: ***good as expected***
Actual Summary: as expected
Sentence: *this is an excellent tea for a breakfast tea or for the afternoon or evening it has a wonderful mellow flavor in the morning i like to brew it with earl grey to create a nice smooth blend it is a great way to start the day*
Predicted Summary: *great product tea*
Actual Summary: wonderful anytime tea

5. Results and Discussion

We compare our results with state-of-the-art abstractive summarization methods.

Table 1: Analyzed Results

Study	*ROUGE-I*	*ROUGE-II*	*ROUGE-L*
[16]	22.71	11.49	21.14
[33]	28.29	14.35	26.38
[50]	31.97	15.23	30.11
AARS (Proposed)	**33.33**	**38.46**	**29.26**

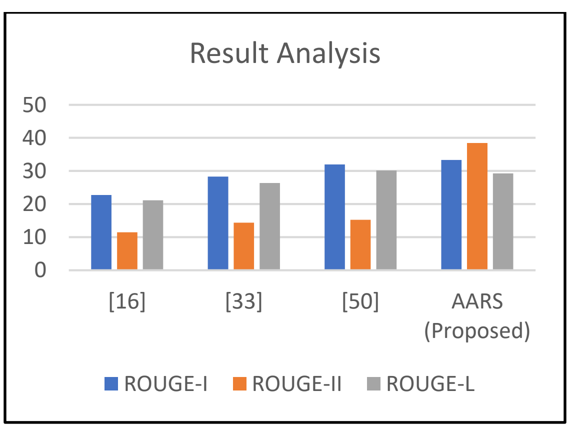

Fig. 5 Result Comparison with base-line studies.

The result indicates that the proposed study have outperformed the base line studies in ROUGE-I, ROUGE-II and ROUGE-L scores, except we have low ROUGE-L score with a ratio of less than 1 percent with respect to [50] due to the reason that the review contents contain slangs, misspelled words, axioms and written by different online users with different writing styles.

6. Conclusion and Future Work

Automatic Summarization provides a smart solution to distill and digest vast amount of information available on Social media platforms and business websites. Aspect based Abstractive review summarization provides a way to give online user a compact aspect-based summary with sentiments, to get a wise decision before making purchase. Among different methods for automatic summarization Bi-GRU encoder-decoder architecture provides compact and accurate review summaries by processing sequences in both directions to get previous as well as future context. This study was intended to generate natural language summaries using Bi-GRU's. We have experimented deep learning-based approach in support of our objectives and achieved an improvement in the ROUGE score with comparison to base line studies. The future work in this domain will be (i) To extend the pre-processing module for spell checking and slang words detection (ii) To use Transformer Networks, to gain benefit of parallel processing for encoder decoder modules. (iii) To extend the abstractive summary generation process for Non-English languages.

Conflict of Interest: Khalid Mahmood, Fazal Masud Kundi and Ghulam Muhammad Kundi and Mahwish Kundi declare that they have no conflict of interest.

References

[1] Luhn, H.P., The automatic creation of literature abstracts. IBM Journal of research and development, 1958. 2(2): p. 159-165.
[2] Edmundson, H.P. and R.E. Wyllys, Automatic abstracting and indexing—survey and recommendations. Communications of the ACM, 1961. 4(5): p. 226-234.
[3] Ganz, F., P. Barnaghi, and F. Carrez, Information abstraction for heterogeneous real world internet data. IEEE Sensors Journal, 2013. 13(10): p. 3793-3805.
[4] Sauper, C. and R. Barzilay. Automatically generating wikipedia articles: A structure-aware approach. in Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1. 2009. Association for Computational Linguistics.
[5] Maybury, M.T., Generating summaries from event data. Information Processing & Management, 1995. 31(5): p. 735-751.

[6] Geng, Z., et al. Post-Processing of Automatic Text Summarization for Domain-Specific Documents. in 2010 International Conference on Communications and Mobile Computing. 2010. IEEE.
[7] Sakai, T. and K. Sparck-Jones. Generic summaries for indexing in information retrieval. in Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. 2001.
[8] Lin, C.-Y. and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. in Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics. 2003.
[9] Alguliyev, R.M., R.M. Aliguliyev, and N.R. Isazade, An unsupervised approach to generating generic summaries of documents. Applied Soft Computing, 2015. 34: p. 236-250.
[10] Krishna, R., S.P. Kumar, and C.S. Reddy, A hybrid method for query based automatic summarization system. International Journal of Computer Applications, 2013. 68(6): p. 39-43.
[11] Manabu, O. and M. Hajime, Query-Biased Summarization Based on Lexical Chaining. Computational Intelligence, 2000. 16(4): p. 578-585.
[12] Bhaskar, P. and S. Bandyopadhyay. A query focused multi document automatic summarization. in Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation. 2010.
[13] Nema, P., et al., Diversity driven attention model for query-based abstractive summarization. arXiv preprint arXiv:1704.08300, 2017.
[14] Hasselqvist, J., N. Helmertz, and M. Kågebäck, Query-based abstractive summarization using neural networks. arXiv preprint arXiv:1712.06100, 2017.
[15] Radev, D., A. Winkel, and M. Topper. Multi document centroid-based text summarization. in ACL 2002. 2002. Citeseer.
[16] Nallapati, R., et al., Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023, 2016.
[17] Tas, O. and F. Kiyani, A SURVEY AUTOMATIC TEXT SUMMARIZATION. PressAcademia Procedia, 2007. 5(1): p. 205-213.
[18] Gallant, S.I. and S.I. Gallant, Neural network learning and expert systems. 1993: MIT press.
[19] Rosenblatt, F., Perceptron simulation experiments. Proceedings of the IRE, 1960. 48(3): p. 301-309.
[20] Rumelhart, D.E., et al., Sequential thought processes in PDP models. Parallel distributed processing: explorations in the microstructures of cognition, 1986. 2: p. 3-57.
[21] Hopfield, J.J., Hopfield network. Scholarpedia, 2007. 2(5): p. 1977.
[22] Hochreiter, S. and J. Schmidhuber, Long short-term memory. Neural computation, 1997. 9(8): p. 1735-1780.
[23] Cho, K., et al., Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.
[24] Deselaers, T., et al. A deep learning approach to machine transliteration. in Proceedings of the Fourth Workshop on Statistical Machine Translation. 2009. Association for Computational Linguistics.
[25] Noda, K., et al., Audio-visual speech recognition using deep learning. Applied Intelligence, 2015. 42(4): p. 722-737.
[26] Wang, F.-Y., et al., Where does AlphaGo go: From church-turing thesis to AlphaGo thesis and beyond. IEEE/CAA Journal of Automatica Sinica, 2016. 3(2): p. 113-120.
[27] Dehghan, A., et al., View independent vehicle make, model and color recognition using convolutional neural network. arXiv preprint arXiv:1702.01721, 2017.
[28] AbdAlmageed, W., et al. Face recognition using deep multi-pose representations. in 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). 2016. IEEE.
[29] Xu, K., et al. Show, attend and tell: Neural image caption generation with visual attention. in International conference on machine learning. 2015.
[30] Gerani, S., et al. Abstractive summarization of product reviews using discourse structure. in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
[31] Sutskever, I., O. Vinyals, and Q.V. Le. Sequence to sequence learning with neural networks. in Advances in neural information processing systems. 2014.
[32] Rush, A.M., S. Chopra, and J. Weston, A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685, 2015.
[33] See, A., P.J. Liu, and C.D. Manning, Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368, 2017.
[34] Chen, S., A General Model for Neural Text Generation from Structured Data. E2E NLG Challenge System Descriptions, 2018.
[35] Yang, M., et al. Aspect and sentiment aware abstractive review summarization. in Proceedings of the 27th international conference on computational linguistics. 2018.
[36] Gao, S., et al. Abstractive Text Summarization by Incorporating Reader Comments. in Proceedings of the AAAI Conference on Artificial Intelligence. 2019.
[37] Bahdanau, D., Neural Machine Translation by Jointly Learning to Align and Translate. Proceedings of the 2015 International Conference on Learning Representation (ICLR), 2015.
[38] Ganesan, K., C. Zhai, and J. Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. in Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010). 2010.
[39] Di Fabbrizio, G., A.J. Stent, and R. Gaizauskas, A Hybrid Approach to Multi-document Summarization of Opinions in Reviews. INLG 2014, 2014: p. 54.
[40] Nallapati, R., B. Xiang, and B. Zhou, Sequence-to-sequence rnns for text summarization. 2016.
[41] Bahdanau, D., et al. End-to-end attention-based large vocabulary speech recognition. in 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). 2016. IEEE.
[42] Gerani, S., G. Carenini, and R.T. Ng, Modeling content and structure for abstractive review summarization. Computer Speech & Language, 2019. 53: p. 302-331.
[43] Puduppully, R., L. Dong, and M. Lapata. Data-to-text generation with content selection and planning. in Proceedings of the AAAI Conference on Artificial Intelligence. 2019.

[44] Speer, R. and J. Lowry-Duda, Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. arXiv preprint arXiv:1704.03560, 2017.
[45] Tsukiyama, D. and J. McAuley, Amazon Fine Food Reviews… wait I don't know what they are reviewing.
[46] Nadkarni, P.M., L. Ohno-Machado, and W.W. Chapman, Natural language processing: an introduction. Journal of the American Medical Informatics Association, 2011. 18(5): p. 544-551.
[47] Liu, Y., et al. Topical word embeddings. in Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015.
[48] Asghar, M.Z., et al., Exploring deep neural networks for rumor detection. Journal of Ambient Intelligence and Humanized Computing, 2019: p. 1-19.
[49] Bahdanau, D., K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
[50] Tian, Y., J. Yu, and J. Jiang. Aspect and opinion aware abstractive review summarization with reinforced hard typed decoder. in Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019.

Khalid Mahmood is a Ph.D. student at Institute of Computing and Information Technology, Gomal University, Dera Ismail Khan, Pakistan. He got his master's in computer sciences (MSCS) degree in Computer from ICIT, Gomal University. His research area are Text Mining, Machine Learning, Multimedia Information Retrieval and Deep Learning Techniques.

Dr. Fazal Masud Kundi is an Assistant Professor at ICIT, Gomal University, Pakistan. He got his master's degree in Computer Science from ICIT and Ph.D. degree in Computer Science from Gomal University, Pakistan . He has more than 50 publications in journals of international repute (JCR and ISI indexed). His research focuses on Sentiment Analysis, Text mining and Machine Learning.

Dr. Ghulam Muhammad Kundi earned master's degree in Public Administration and Political Science. Dr. Kundi did his Ph.D. in management studies (e-Business) in 2009, and Post-Doc fellowship in e-Government from Aix Marseille University, France. He is a full Professor at Faculty of Law and Administrative Studies, Gomal University, Pakistan. Dr. Kundi has more than 70 publications (Scopus, ERA and ISI indexed) journals. His area of interest is digital management/digital marketing and expertise in quality assurance and accreditation

Dr. Mahwish Kundi is working as Assistant Professor in Computer Science Department at Abdul Wali Khan University Mardan. She got his master's degree in computer science (MSCS) from ICIT in 2011 and Ph.D. (Software Engineering) from University of Leicester in 2019. Her area of interest is Requirements Engineering and Software Quality Metrics. Before joining Abdul Wali Khan University, she worked as a Lecturer in Institute of Computing and Information Technology Gomal University D.I.Khan (2007-2012).