Sentic Computing for Aspect-based Opinion Summarization using Multi-head Attention with Feature Pooled Pointer Generator Network

Akshi Kumar[1] · **Simran Seth**[1] · **Shivam Gupta**[1] · **Shivam Maini**[1]

Conflict of Interest Statement

The authors certify that there is no conflict of interest in the subject matter discussed in this manuscript.

Author Identifying Statement

Akshi Kumar: Conceptualization, Methodology, Validation, Writing, Reviewing and Editing

Simran Seth: Methodology, Implementation, Validation, Writing

Shivam Gupta: Methodology, Implementation, Validation, Writing

Shivam Maini: Implementation, Writing

Akshi Kumar
E-mail: akshikumar@dce.ac.in

[1]Department of Computer Science & Engineering,
Delhi Technological University, Delhi 110042, India

Abstract

Background/Introduction: Neural sequence to sequence models have achieved superlative performance in summarizing text, owing to their adaptive learning and generalization capabilities. But they tend to generate generic and simplistic summaries that under-represent the opinion-sensitive aspects of the document. They use domain-independent and uniform language modelling techniques to summarize the text. Therefore, they fail to capture salient sentiment-oriented information from the source text. Additionally, the sequence to sequence models are prone to test-train discrepancy (exposure-bias) arising from the differential summary decoding processes in the training and testing phases. The models use ground truth summary words in the decoder training phase and predicted outputs in the testing phase. This inconsistency leads to error accumulation and substandard performance. Motivated by these drawbacks, a cognitive aspect-based opinion summarizer, **f**eature **p**ooled **p**ointer **g**enerator **n**etwork (FP2GN) is proposed which selectively attends to thematic and contextual cues to generate sentiment-aware review summaries.

Methods: This study augments pointer generator framework with opinion feature extraction, feature pooling and mutual attention mechanism to tackle the problem of opinion summarization. The proposed model FP2GN identifies the aspect terms in review text using sentic computing (SenticNet 5 and concept frequency inverse opinion frequency) and statistical feature engineering. The aspect terms present in the source text are encoded into context embeddings using weighted average feature pooling. These embeddings are processed in a pointer-generator framework inspired stacked Bi-LSTM encoder-decoder model with multi-head self-attention. The decoder system uses temporal and mutual attention mechanisms to ensure the appropriate representation of input-sequence at each decoding step. The study also proffers the use of *teacher forcing ratio* to curtail the exposure-bias related error-accumulation.

Results: The model achieves ROUGE-1 score of 86.04% and ROUGE-L score of 88.51% on Amazon Fine Foods dataset. The proposed weighted average pooling technique is compared with other cognitive aspect-fusion methods and an average gain of 2% is observed. The results ascertain that FP2GN is a space and time-efficient opinion summarizer that generates state-of-the-art sentiment-aware summaries.

Conclusion: The proposed model reinforces pointer generator network architecture with opinion feature extraction, feature pooling and mutual attention mechanism to generate human-readable opinion summaries. Empirical analysis substantiates that the proposed model is better than the baseline opinion summarizers.

Keywords Pointer generator network · opinion summarization · self-attention · sequence-sequence · sentic computing

1 Introduction

Web 2.0 has given way to social networking renaissance marked with an increase in the amount of opinionated social media data. This increase can be attributed to the growth in the number of individuals using social media, review sites, and other such platforms to share their beliefs and opinions with the world [1]. The ever-widening

flow of opinion-sensitive data [2–4] has essentially changed the face of modern-day consumer relationship management and customer-enterprise alliance [5]. Market stakeholders ranging from industrialists to political decision makers keep tabs on social-media sentiment to foster trust and cooperation. The comments that are posted on various online platforms examine products, people, and services, based on various aspects of their performance. Research shows that such critical comments influence up to 82% of all (purchase, election etc.) decisions made by the users [6]. However, the sheer size of the available information makes it impossible to manually read the opinion-sensitive texts before forming an absolute opinion. The surge of available data has resulted in difficulties around kneading data into a structured and usable form. It is thus imperative to upgrade data filtering systems to retrieve key content that captures salient information from the source text. Automatic text summarization involves condensation of source articles into a concise version that reflects its central theme comprehensively. It finds its application in a myriad of interesting real-life use-cases like social media analytics, question-answering automation, news condensation and customer relationship management [7]. Broadly, two methods of text summarization are discussed in the literature, namely extractive text summarization and abstractive text summarization [8]. Extractive summarization involves selection of contextually important sentences from the original text, and their placement into the summary [7]. The main feature of extractive text summarization is that it does not modify the sentences present in the reference text. On the other hand, abstractive summarization aims to produce a holistic summary by creating original sentences. It enables generation of high-quality summary by incorporating sophisticated methodologies like paraphrasing, generalization, and context-adherence.

Researchers have succeeded in capturing thematic information from text using conventional summarization techniques, but mining opinion-sensitive information remains a challenge. Therefore, the development of domain-independent, subject-consistent and interpretable artificial intelligence models to summarize user opinions is indispensable. Opinion summarization is the generation of holistic review summary that efficiently captures the idea and sentiment of source text [9]. It is different from conventional text summarization since the phrases that are factually instructive may not always represent the opinion-oriented state of affairs. Opinion summarization provides a coherent and concise representation of opinionated text. It helps in the analysis of the emotional indications that affect the final state of events, and hence finds its application in the domains of business intelligence and social media monitoring [10].

Consequently, sentic computing [11–15], a novel technique for opinion mining and sentiment analysis, is introduced. It takes a holistic approach to tackle the suitcase research problem of sentiment analysis by analysing the explicit and implicit expression of human language. It outperforms existing opinion retrieval and analysis methodologies by exploiting computational linguistics and social sciences to better interpret, identify and process user opinions. Sentic computing finds its applications in a multitude of domains like mathematics, healthcare, social media marketing, psychology, sociology and ethics.

The recent success of neural sequence to sequence (seq2seq) models (Figure 1) in sentic computing tasks like statistical machine translation, named entity recog-

nition, sentiment analysis and sarcasm detection has inspired researchers to build seq2seq framework-driven text summarizers [16]. These models use recurrent neural networks (RNN) to freely read, encode and generate text for accurate summarization. Although these systems are promising, they produce unnatural, repetitive and factually inconsistent summaries. Also, these systems cannot effectively handle the problem of out of vocabulary (OOV) words. These shortcomings have given way to temporal attention, pointer generator network, conditional random forest (CRF) and hierarchical opinion summarization systems.

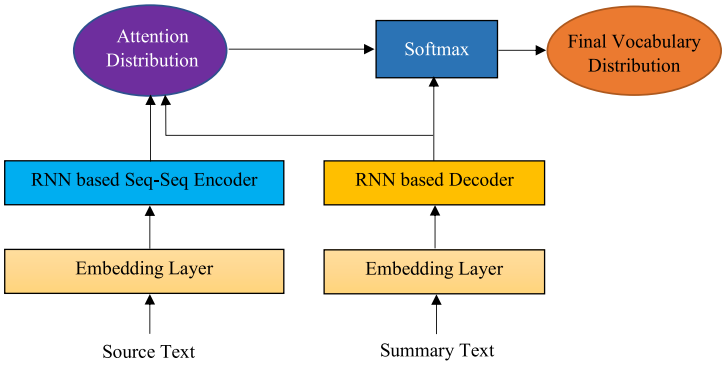

Fig. 1 Schematic Structure of seq2seq system

Concurrently, it has been observed that attention-based seq2seq models fail to produce sentiment-aware summaries. They show a tendency to generate highly generic summaries containing frequent phrases from the source text [9]. Pertinent research methods [16, 17] use summarization models with high generalization capabilities that tend to miss or under-represent the salient features of reference-text. They fail to encapsulate aspect and sentiment-oriented details, which are of high value to the stakeholders. Besides, they are partly autoregressive in nature and exhibit "exposure bias". Exposure bias refers to the train-test inconsistency arising from autoregressive generation models that use ground truth sequences during training and predicted outputs during testing. Therefore, seq2seq systems are not accustomed to incorporate their own predictions for the task of summarization. This discrepancy results in error accumulation and poor performance.

The two-fold need to firstly curtail the train-test discrepancy arising from the autoregressive nature of neural seq2seq models and secondly to optimize the aspect-based opinion coverage in generated summary, has inspired the premise of **F**eature **P**ooled **P**ointer **G**enerator **N**etwork (FP2GN). This work proposes a cognitive model based on the ensemble application of pointer generator network and sentic computing to generate context adhering and factually consistent summaries with a high degree of sentiment-coverage. Cognitive computing is an iterative, interactive, adaptive, stateful and contextual methodology that aids information retrieval and decision analytics. It promotes the application of multiple intelligent technologies like machine learning, deep learning, artificial intelligence, natural language processing and pattern recogni-

tion to solve problems requiring real-time intelligence. This study explores the utility of pointer generator framework based Bi-LSTM encoder decoder model to categorize, classify and remember input information for remarkable opinion summarization. We aim to mimic human cognitive capabilities to comprehend contextual cues and generate natural and human-readable summaries, using aspect-term categorization and fusion, pointer generator network and interactive mutual attention mechanism.

Pointer generator network efficiently handles the out of vocabulary (OOV) problem of seq2seq models. It facilitates the choice between copying a word from input sequence and generating a word out of the document vocabulary. This study proposes aspect extraction and aspect-fused context representation for enhancing the opinion coverage in abstractive summary. Aspect terms (or opinion features) are representative of phrase-level topics that have an associated opinion polarity. Aspect extraction is a highly subjective and context-dependent task [9] that requires comprehending thematic cues from the reference text. To this end, sentic computing is used to couple common-sense reasoning with opinion retrieval, affective computing and emotion categorization to effectively classify aspect terms (Figure 2). In this work, aspect terms are identified and incorporated in the neural encoding of the reference-text to ensure their adequate representation in generated summary.

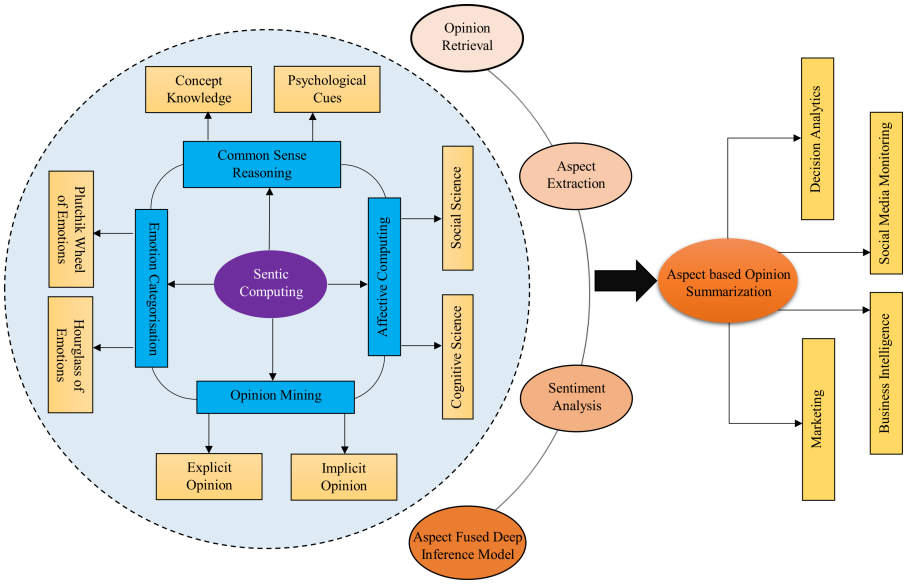

Fig. 2 Application of sentic computing to opinion summarization

The proposed FP2GN model proffers a multi-head mutual attention-based pointer generator network with opinion feature extraction and pooling. Prospective opinion feature terms (aspect terms) are first identified using contextual information (context-similarity) and feature engineering techniques like TextRank [18], RAKE [19] and concept frequency-inverse opinion frequency scores (CF-IOF) [20]. SenticNet 5 [21]

is used to attach opinion polarities to aspect-terms for CF-IOF analysis. The opinion feature set is then used for sequence labelling of reference-text to distinguish aspect terms from other words. Skip-Gram Word2Vec, a pretrained embedding model is used to generate feature embedding matrix for the tagged input sequence. Skip-Gram Word2Vec is a semi-supervised training technique that uses the neighboring words for theme comprehension and label assignment. The feature embeddings of opinion features are weighted average pooled with all the other words to obtain aspect-fused input-sequence. The feature embeddings so obtained are fed as input to the stacked Bi-LSTM encoder that can efficiently capture the past and future context information in reference-text. The Bi-LSTM output vector is processed in the multi-head self-attention layer to obtain encoder context vector. Attention layer ensures that different semantic subspaces of input-sequence are adequately represented in generated summary. The Bi-LSTM output vector also initializes the pointer generator-based LSTM decoder. The decoder used in this model is partly autoregressive as it uses *teacher forcing ratio* to incorporate both the ground truth context and the predicted summary information for opinion summary generation. LSTM decoder probabilistically chooses between the ground truth summary word and the previously generated word for enhanced model training. LSTM output vector is passed into the temporal attention layer, mutual soft-attention layer and the softmax layer. The temporal self-attention layer handles the repetition problem of conventional seq2seq models by incorporating information about the previously decoded sequence. Mutual soft-attention layer uses the soft-attention mechanism to calculate the inter encoder-decoder context vector and ensures appropriate representation of reference(input) context at each decoding step. The softmax layer uses the pointer-generator framework to eliminate the out of vocabulary problem. It chooses between copying words from input sequence and generating words from pre-defined vocabulary to obtain the final vocabulary distribution. The key contributions of this paper can be summarized as:

- We leverage sentic computing techniques like CF-IOF and SenticNet 5 for opinion feature classification. The phrase-level opinion features are identified and average pooled with other context words present in reference-text to exploit their neighborhood semantic properties. As a result, human-readable, specialized and factually consistent summaries are obtained.
- Three kinds of attentions namely multi-head encoder-self attention, decoder temporal self-attention and inter encoder-decoder mutual attention are explored for repetition avoidance and selective representation of thematic and contextual information while decoding.
- We propose the use of teacher forcing probability to deal with the issue of exposure bias. Teacher forcing probability enables FP2GN to incorporate and learn from the decoded outputs.
- The study explores the utility of the proposed sentic computing-based opinion summarization technique in the field of Business Intelligence. Amazon fine foods dataset is used to validate the efficacy of FP2GN for mining relevant experiential information.

The rest of the paper is organised as follows: section 2 discusses the related research within the domain of opinion summarization. Section 3 elucidates the details of the proposed FP2GN model and all the component modules. Working examples from dataset are illustrated in Section 4. Section 5 details the baselines, validation data, experimentation and the results. Further, conclusion and future works are discussed in section 6. The frequently used abbreviations are tabulated in Table 1.

Table 1 List of abbreviations

Abbreviation	**Full Form**
FP2GN	Feature Pooled Pointer Generator Network
LSTM	Long Short Term Memory
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
PGC	Pointer Generator Network with Coverage
PSO	Particle Swarm Optimization
OOV	Out of Vocabulary
AOS	Aspect-based Opinion Summarization
CF-IOF	Concept Frequency-Inverse Opinion Frequency
CNN	Convulational Neural Networks
TC	Target Concatenation
ABS	Abstractive Summarizer
RNN	Recurrent Neural Network
NLP	Natural Language Processing

2 Related Work

The growing significance of content posted on social-media and feedback platforms has intensified researchers' interest to rationally mine opinionated data for social media analytics and consumer-enterprise relationship management [10]. Researchers' worldwide are developing methodologies for automating the process of text and opinion summarization to tap user-opinion within the big-pool of social media data. Pertinent literature discusses various extractive and abstractive summarization models. Other emerging areas in the field of opinion summarization include aspect-fused [22], query-focused [23], real-time [24] and contrastive [25] summarization techniques. Literature is well equipped with primary and secondary studies on state-of-the-art opinion summarization techniques [7, 10].

Primary studies on opinion summarization majorly focus on the use of extractive techniques. Researchers have developed a variety of extractive text summarization models. Broadly, all these methodologies fall under one of the following categories: term frequency-inverse document frequency methods, cluster-based methods, graph theoretic approaches, machine learning approaches, automatic T-S (Triangle Similarity) based on fuzzy logic, LSA (Latent Semantic Analysis) method, text summarization with neural networks and query based extractive text summarization [8]. Manjula et. al [26] summarized text in the Hindi language by constructing a graph representing the semantics of a paragraph, and then isolating the sub-graph containing the

'key' summary. Qasem et al. [27] presented hybrid single document summarization model ASDKGA (Arabic Text Summarization Using Domain Knowledge and Genetic Algorithms). They proposed the use of domain knowledge, statistical features and genetic algorithms for sentence-scoring in Arabic political text. Nallapati et. al [28] built SummaRuNNer, an RNN based model. This model learns abstract features like 'context', 'theme' and 'sentiment' from human written summaries to generate holistic extractive summaries. Rodríguez et. al [29] employed a decomposition-based approach. The artificial bee colony algorithm was deployed in a multi-objective fashion to create text summaries. Mudasir et. al [30] showed the importance of using semantic features like word embeddings in enhancing the summarization quality. Rajangam et. al [31] proposed a novel cognitive model that uses hierarchical human memory model. It used knowledge-based event-index to create text summaries.

Though extractive summarization techniques achieve high ROUGE-scores, they lack continuity, context coherence and human readability. These shortcomings open avenues for the use of natural language generation techniques to perform opinion summarization. Abstractive summarization is broadly divided into two categories namely structure based and semantic summarization. Structure based techniques may fall into one of the following categories: tree-based, template-based, ontology-based, lead and body phrase and rule-based methods. The semantic based techniques include the following methodologies: multimodal semantic model, information item- based method and semantic graph-based method [10]. Tsutomu et. al [32] used a tree structure in which nodes represented the discourse of the matter. The tree is trimmed in order to summarize the text. See et. al [16] proposed PGC (Pointer Generator with Coverage) which uses pointer generator network with coverage mechanism to deal with the repetition and out of vocabulary problems in baseline abstractive models. Nallapati et al. [33] proposed ABS, an attentional encoder decoder recurrent neural network for abstractive text summarization. Zheng et. al [34] used segmental encoder-decoder network learning. The proposed network is hierarchical and adaptive in nature and summarizes text in an abstractive manner. Moirangthem et. al [35] used deep, hierarchical and temporal pointer generator networks to handle long sequences. Adelia et. al [36] used bidirectional gated recurrent units (GRU) so that the sentences in the generated summary are influenced by surrounding words. Paulus et. al [37] proposed DeepRL (Deep Reinforcement Learning), which uses neural networks with reinforcement learning and a unique intra-attention mechanism to efficiently handle long documents. GANsum (Generative Adversarial Network for summarization) proposed by Liu et. al [38], uses simultaneously trained generative and discriminatory models to generate human-readable summaries.

The last decade has witnessed the growth of sentic computing [11] as a novel perspective to computational linguistics. Pioneer studies in the field focussed on the use of common-sense, AI, emotion categorization and affective computation for sentiment analysis subtasks like sarcasm detection, named entity recognition, multi-modal and multi-lingual sentiment classification, and aspect extraction [11, 14]. Dragoni et al. [39] used open information extraction strategies for real-time aspect-based sentiment analysis. Opinion aggregation based sentic computing technique is used for aspect extraction and polarity assignment. Ma et al. [4] proposed the use of a two-step attentive neural architecture along with Sentic LSTM to improve aspect catego-

rization and sentiment assignment. Contemporary studies explore the application of sentic computing methodologies to real-life use cases like social media monitoring [39], behaviour analytics [40], healthcare [41], finance and business intelligence [42].

Simultaneously, the impact of thematic, contextual and emotional cues has been explored in relevant literature studies on aspect-based opinion summarization. Yang et al [9] developed the sentiment-aware model MARS (Multi-factor attention fusion network for aspect/sentiment-aware Abstractive Review Summarization). An interactive attention mechanism is used to learn the representation of aspect, sentiment and context words to summarize reviews. Deng et. al [43] used sequence to sequence model to integrate keyword information into the generated summaries. Article's key content is captured using attention mechanism. Pan et. al [44] developed a model that captures scenic information. It uses an RNN-based encoder-decoder system with external attention. Peng Wu et. al [45] proposed the use of Ortony-Clore-Collins (OCC) model of emotional analysis along with convolutional neural network (CNN) for opinion summarization of micro blog texts. Zhou et. al [46] used query-based opinion mining and summarization techniques for international e-commerce reviews. They employed multi-granularity opinion mining to study the difference between the online shopping behaviours of America and China. Abdi et. al [47] proposed a lexica-based model to summarize documents in a query-based sentiment-oriented manner.

Our model FP2GN uses multi-head mutual attention-based pointer generator network with aspect-fusion and opinion feature pooling. It improves the performance of baseline attention-based encoder-decoder systems by incorporating aspect-oriented information into the encoded reference-text. The research explores and validates the use of CNN-like pooling techniques to integrate contextual cues into text-embeddings. Three types of attentions namely 'multi-head encoder self-attention', 'decoder temporal attention' and 'inter encoder-decoder mutual attention' are used to adequately represent sentiment information present in input-sequence. Further, improved teacher forcing algorithm is proposed to contain error-accumulation due to exposure-bias. The proposed **f**eature **p**ooled **p**ointer **g**enerator **n**etwork (FP2GN) is explained in the next section.

3 The Proposed FP2GN Model

The proposed **F**eature **P**ooled **P**ointer **G**enerator **N**etwork (FP2GN) augments the strength of pointer generator network [16] with opinion feature extraction, feature pooling and mutual attention mechanism to tackle the problem of opinion summarization. The coordination of opinion features (aspect terms) and context words in the reference text is exploited to yield opinion-sensitive summaries. FP2GN model mainly consists of 4 modules namely (1) opinion feature extraction (2) opinion feature mapping (3) feature pooled stacked Bi-LSTM based encoder and (4) LSTM based decoder. Figure 3 shows the architectural design of the feature-pooled pointer generator network.

In the following subsections the details of each of these modules will be expounded.

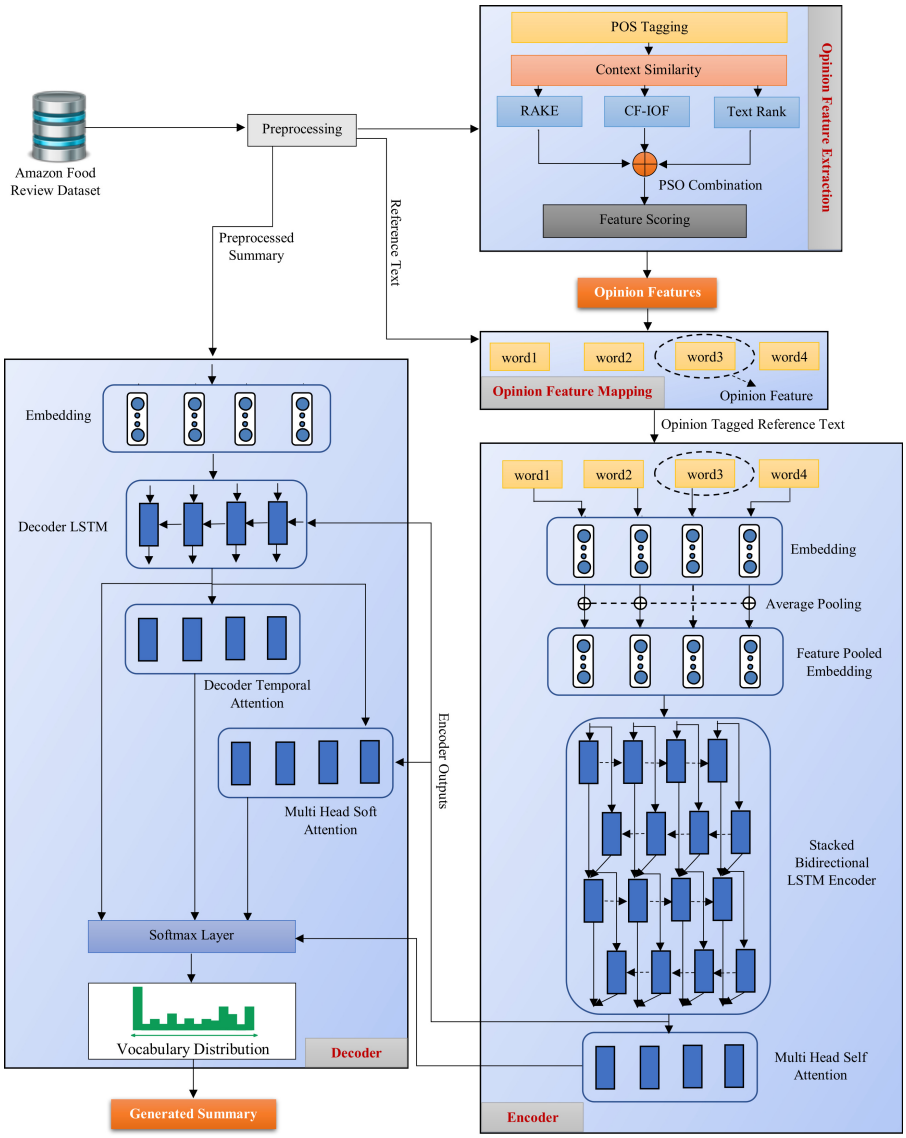

Fig. 3 Architecture of FP2GN

3.1 Dataset acquisition

Amazon fine-food dataset [48] has been used for the task of opinion summarization in this paper. The dataset contains 568,454 reviews for 74,258 products spanning over a period of 10 years. It contains information about the users, food products and associated ratings, plain text review and summary. The dataset has widely been used for research works in the fields of recommendation, sentiment analysis, summariza-

tion, and topic modelling. Each tuple in the dataset mainly contains Product Identifier, User Identifier, Score, Time, Summary, Review Text etc. Table 2 contains the detailed description of the data attributes.

Table 2 Attributes and Descriptions

Column Name	Description
Product Identifier	Uniquely identifies product
User Identifier	Uniquely identifies user
Profile Name	User's profile name
Helpful Votes	Count of helpful votes for the review
Total Votes	Total number of votes for the review
Rating	Rating ranging from 1 to 5
Time	Review Timestamp
Summary	Summary of the review
Review Text	Text of the review

3.2 Data Preprocessing

After the acquisition of data, data pre-processing is performed. This step is necessary to transform the data into a form suitable for opinion feature extraction and opinion summarization. The process involves the following steps:

1. **Tokenization –** The reference text and associated summary is tokenized using the Python Natural Language Toolkit (NLTK) to filter individual elements (words, symbols, punctuation characters) called tokens.
   For example, the summary, "The food was delicious and worth reordering!!" is tokenized to yield the following tokens:
   ['The', 'food', 'was', 'delicious', 'and', 'worth','reordering', '!', '!']
2. Next, we remove the tokens that categorize as one of the following: numeric text, empty text, URL, mention, hash tag, stop word and punctuation. Table 3 briefly describes the categories of removed tokens.

Table 3 Categories of removed tokens

Category	Examples
Numeric Text	1234, 0465, 665, 4647
Empty Text	" "
URL	https://www.wikipedia.org/, http://dtu.ac.in/
Mention	@Shivam, @Simran
Hash Tag	#tasty, #delicious
Stop word	is, an, the
Punctuation	, ! @ () []

3. **Lemmatization:** The final pre-processing step is lemmatization. This step involves transforming all tokens to their lemmas or dictionary form. Table 4 shows the dictionary form (lemma) of a few English words.

Table 4 Dictionary forms of some English word.

Word	Lemma
studying	study
dying	die
complaining	complaint
cooking	cook
listening	listen
stupidity	stupid

The following example depicts how a given reference-text is pre-processed. Consider the text: "I ordered a pizza yesterday. I have to say that it was the greatest pizza I have eaten in recent times!! It was filled with cheese and delicious toppings."

(a) The tokens generated are:
['I', 'ordered', 'a', 'pizza', 'yesterday', '.', 'I', 'have', 'to', 'say', 'that', 'it', 'was', 'the', 'greatest', 'pizza', 'I', 'have', 'eaten', 'in', 'recent', 'times', '!', '!', 'It', 'was', 'filled', 'with', 'cheese', 'and', 'delicious', 'toppings', '.']

(b) Removal of unnecessary tokens (described in step 2):
['ordered', 'pizza', 'yesterday', 'say', 'greatest', 'eaten', 'recent', 'times', 'filled', 'cheese', 'delicious', 'toppings']

(c) After lemmatization, the final tokens used for opinion feature extraction and opinion summarization are:
['order', 'pizza', 'yesterday', 'say', 'great', 'eat', 'recent', 'time', 'fill', 'cheese', 'delicious', 'topping']

3.3 Opinion Feature Extraction

Sentiment analysis is an exemplary text analysis technique whereby the sentiment polarity of opinionated social media text is classified as positive, negative, and neutral [9]. Sentiment analysis tackles various natural language processing subtasks including sarcasm detection, named entity recognition, multi-modal and multi-lingual sentiment classification, and aspect extraction [2]. Aspect extraction has emerged as a classic NLP subtask that can highly boost the accuracy of sentiment classification. Aspect-oriented sentiment analysis associates the phrase-level opinion polarity with specific aspects (opinion features) rather than the complete text entity. Aspect terms, also referred to as opinion features are the n-gram phrases in a sentence that have an associated opinion polarity. For example, in the sentence "The food tastes really good, although the presentation is quite disappointing", the opinion polarity towards the aspect "food" is positive whereas, the sentiment expressed towards the aspect

"presentation" is negative. Thus, sentence level opinion aggregation may lead to factual discrepancies. Aspect extraction finds its application in aspect-oriented opinion summarization (AOS). Standard AOS involves aspect identification and sentiment classification. Given a set of product reviews, an AOS system labels the opinion features discussed in the reviews and predicts reviewers' sentiments towards those aspects. Aspect extraction and aspect-based sentiment classification lead to fine-grained depiction of popular opinion about specific products.

In our model, we focus on the extraction of unigram noun features. For example, in the sentence "my dog loved the biscuits. I am glad I purchased them", "biscuits" is the unigram noun feature towards which the reviewer shows attitude. The opinion feature extraction module used in FP2GN is inspired from the candidate feature mining technique proposed by S. Hu et al. [49]. The pre-processed word lemmas (obtained from the pre-processing module) are POS (part of speech) tagged to filter out the noun features. The unigram noun lemmas are then scored based on their spaCy[1] similarity with the root context("food"). The words with similarity $> \delta$ are further processed to obtain highly relevant opinion features ($\delta = 0.4$ empirically). The premise of using context similarity derives its foundation from AffectiveSpace[14] concept similarity [20], whereby 'eigenmoods' are analysed using principal coordinates in the conceptual vector space. 'eigenmoods' describe common sense concepts and emotions in a vectorized fashion, whereby the vector coordinates describe sentiment (mood) using the axes of AffectiveSpace. For example, the most significant eigenmood, e_0, is representative of positive affective polarity. Therefore, the larger the concept's vector component aligns with the e_0 direction, the more affectively positive it is likely to be. FP2GN tends to analyse the context-similarity of aspects with respect to the core-context ('food' in this case) using spaCy cosine similarity. Subsequently, concept frequency-inverse document frequency is used to analyse the associated sentiments and aspect importance.

The RAKE [19], TextRank [18] and CF-IOF (concept frequency inverse opinion frequency) [20] scores are evaluated and linearly combined using particle swarm optimization to obtain final feature scores. PSO assigns initial random weights to the three metrics and updates them iteratively to optimize the ROUGE-L score. Hence, the coefficients of linear combination are the learnable parameters that are adjusted to maximize the cost function, i.e. ROUGE-L score. The aspect terms with considerable feature scores are characterized as opinion features. RAKE ensures that interesting features with appreciable co-occurrence with opinion words are highly scored. TextRank identifies and scores the most influencing (important) features. CF-IOF score leverages fine-grained sentiment analysis to ensure that the opinion feature set contains only opinion devising domain-dependent features. The fine-grained sentiment analysis involves sentiment scoring indicative of strong/weak sentiment intensities associated with the subtleties of human language. It breaks down the predicted sentiment into five discrete classes, namely, highly negative, negative, neutral, positive and highly positive. The fine-grained polarity values are attached to each filtered opinion feature using SenticNet 5 [21] and fastText[2] as illustrated in Figure 4. The mathe-

[1] https://spacy.io/models/en

[2] https://github.com/facebookresearch/fastText

matical equations for calculating the CF-IOF score for feature i is given in equation 1.

$$(\text{CF-IOF})_i = \sum_j \frac{n_{ij}}{\sum_k n_{kj}} \times \log\left(\frac{5}{|\{r : o_i \in r\}|}\right) \tag{1}$$

where,
n_{ij} = number of reviews containing feature i with opinion level j
r = reviews
o_i = feature i
$|\{r : o_i \in r\}|$ = number of reviews containing feature i

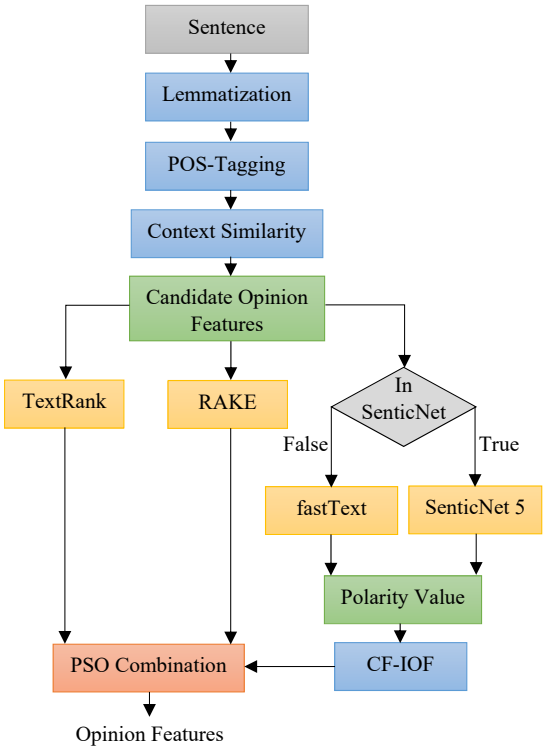

Fig. 4 Opinion feature extraction

SenticNet 5 is a collection of 10,000 n-gram entries where $n \in [1,5]$. It helps in assigning polarity values to concepts at each semantic level according to the Hourglass of Emotions [20]. SenticNet 5 outputs floating polarity values between -1 and +1 (where -1 is extreme negativity and +1 is extreme positivity). The floating polarity values are grouped into bins of width 0.4, where each bin corresponds to an integral polarity $\in \{1,5\}$. For instance, the bin [-1, -0.6] corresponds to the integral polarity of 1. FastText is used to assign polarity values to the filtered features that are not present in SenticNet 5. FastText uses the internal representational encoding

of words for fine-grained sentiment analysis. Since our model deals with real product reviews that capture complex human emotions, fastText provides the desired adaptability without compromising the space and time efficiency.

A balanced combination of the aforementioned three scores leads to optimum opinion feature extraction results. Some examples of the extracted unigram features, and their corresponding context similarity, CF-IOF, RAKE and TextRank scores are illustrated in table 5. The proposed opinion feature extraction performs better than the other cognitive computation inspired opinion aggregation-based models [39] because of the context-similarity and adaptive polarity assignment techniques. Our model bridges the gap between statistical NLP (TextRank and RAKE) and cognitive-inspired sentic computing techniques (CF-IOF) to enhance the efficacy of aspect-term extraction.

Table 5 Opinion Feature Scores

Feature	**Similarity with "food"**	**Rake Score**	**TextRank Score**	**CF-IOF Score**
taste	0.563	0.662	0.586	0.426
coffee	0.572	0.412	0.338	0.361
tea	0.491	0.379	0.315	0.352
chocolate	0.486	0.428	0.381	0.379
quality	0.401	0.685	0.614	0.574

3.4 Opinion Feature Mapping

The pre-processed reference text and the opinion feature set are fed as input to the opinion feature mapping module for sequence labelling of the input text. The opinion features are marked for pooled embedding [Sec. 3.5] using the BIO tagging scheme [50]. Each word w_i in reference text is tagged as $t_i \in \{B, I, O\}$ (B: Beginning, I: Inside, O: Other), where 'B' denotes the beginning of opinion feature in reference text, 'I' indicates the words that are part of opinion feature(other than the first word) and 'O' signifies all the other words. For example, the sentence, "My dog loved the cream biscuits" is sequence labelled as "My**/O** dog**/O** loved**/O** the**/O** cream**/B** biscuits**/I**" using the BIO tagging scheme. Here "cream biscuits" is the opinion feature which is sequence labelled as "cream**/B** biscuits**/I**".

3.5 Feature-Pooled Stacked Bi-LSTM based Encoder

The encoder takes feature tagged word sequence of length n, $X = \{x_i \mid i \in [1, n]\}$, as input and produces its encoded opinion-oriented context representation. It consists of the following layers:

1. *Embedding Layer:* Embeddings map discrete categorical variables to lower dimension, learned, continuous real valued vectors. Neural network embeddings

are useful for natural language processing tasks since they contextually represent categories in transformed lower dimension space [51]. In this model, Skip-Gram Word2Vec method has been used for generating word embeddings, $W = \{w_i \mid i \in [1,n]\}$ with a dimension(D) size of 256 and a batch size of 50. The BIO tags associated with the input word sequence are concatenated with word embeddings to distinguish opinion features from other words present in the sequence.

2. *Feature-Pooled Embedding Layer:* Most of the neural network based seq2seq models use recurrent neural networks like LSTM [52] to model the context of sentences. But the biggest shortcoming associated with RNN based context modelling is that it fails to distinguish between aspect terms and other words. To overcome this, we propose feature pooled embedding layer that manoeuvres the semantic neighborhood properties of text using weighted average pooling. FP2GN uses weighted average pooling instead of average pooling to learn the relative importance of words in addition to the opinion-level relevance and context coherence. Opinion Feature tagged word embeddings (words with tags $\in$ B,I) are average pooled with embeddings at all other positions to incorporate target information into the encoded sequence. Equation 2 is used to calculate feature pooled word embeddings:

$$fp_i = \left(\frac{W_i \times w_i + \sum_{j \in OF} W_j \times w_j}{1 + |OF|} \right) \tag{2}$$

where,
$fp_i =$ feature pooled embedding for word at position i
$w_i =$ word embedding for word at position i
$OF =$ opinion feature in sentence i.e. words with tags $\in$ B,I
$|OF| =$ number of words categorized as opinion feature in sentence
$W_i =$ learnable weight for word embedding at position i

3. *Stacked Bi-LSTM layer:* The long short-term memory (LSTM) is a special variant of RNN that can proficiently model long-term dependencies. LSTMs make use of three control gates to preserve information pertaining to the input sequence that has been processed by the network. The first forget gate determines the amount of information to be preserved from previous cell state(c_{t-1}). The second input gate regulates the extent of new information to be saved into the current cell state(c_t) from the input(x_t). The third output gate determines the amount of current cell state(c_t) information to be passed onto the output value(h_t). Equations 3 to 7 illustrate the involved calculations:
Input gates:

$$i_t = \sigma(W_i.[x_t, h_{t-1}] + b_i) \tag{3}$$

Forget gates:

$$f_t = \sigma(W_f.[x_t, h_{t-1}] + b_f) \tag{4}$$

Output gates:

$$o_t = \sigma(W_o.[x_t, h_{t-1}] + b_o) \tag{5}$$

Cell states:

$$c_t = f_t \times c_{t-1} + i_t \times \tanh(W_c.[x_t, h_{t-1}] + b_c) \tag{6}$$

Cell outputs:

$$h_t = o_t \times \tanh(c_t) \tag{7}$$

where,
σ denotes the logistic sigmoid function, x_t indicates embedding at the t^{th} position of the sentence, h_t denotes the hidden state, W terms represent weight matrices (e.g. W_i represents the input gate weight matrix) and b terms represent the bias vectors (e.g., b_i represents the input gate bias vector) for the three gates.
LSTM is an efficient technique that can be used to perform various natural language processing tasks like machine translation, sentence completion etc. But conventional LSTMs can only model the past context of sentences. For enhanced efficacy of aspect-oriented opinion summarization, the future context must also be captured by the encoder system. Hence, Bi-LSTM (bidirectional long short-term memory) is used to obtain word features that capture both the previous and future context relations in input sequence. A Bi-LSTM processes the input sequence $X = \{x_i | i \in [1,n]\}$ in the forward direction (from x_1 to x_n) to obtain forward hidden sequence $\overrightarrow{H} = \{\overrightarrow{h}_i | i \in [1,n]\}$, as well as in backward direction (from x_n to x_1) to obtain backward hidden sequence $\overleftarrow{H} = \{\overleftarrow{h}_i | i \in [1,n]\}$. The forward and backward hidden sequences are concatenated to obtain the final output sequence (y_t). The detailed calculations involved are presented in Equations 8 to 10.

$$\overrightarrow{h_t} = \sigma(W_{\overrightarrow{h}}.[x_t, \overrightarrow{h}_{t-1}] + b_{\overrightarrow{h}}]) \tag{8}$$

$$\overleftarrow{h_t} = \sigma(W_{\overrightarrow{h}}.[x_t, \overleftarrow{h}_{t+1}] + b_{\overrightarrow{h}}]) \tag{9}$$

$$y_t = W_y.[\overrightarrow{h}_t, \overleftarrow{h}_t] + b_y \tag{10}$$

where,
$y = (y_1, y_2, \ldots y_t \ldots, y_n)$ is the output sequence and W terms represent the weight matrices.
Literature has sufficient research evidence supporting the hypothesis that deep hierarchical neural networks have better efficacy compared to their shallow counterparts [53]. Therefore, a stacked Bi-LSTM network has been defined wherein the Bi-LSTM outputs from lower layers (y_t^*) are fed into the upper layers as input. It sequentially processes the input sequence to extract important information for enhanced summarization results. Our model uses a 3-stacked Bi-LSTM for encoding the input sequence. Structure of stacked Bi-LSTM is presented in Figure 5.

4. *Multi-Head Self Attention Layer:* Attention layer can be defined as mapping a query and a set of key-value pairs to an output [17]. Final attention is computed as a linear combination of values, where the coefficients of combination

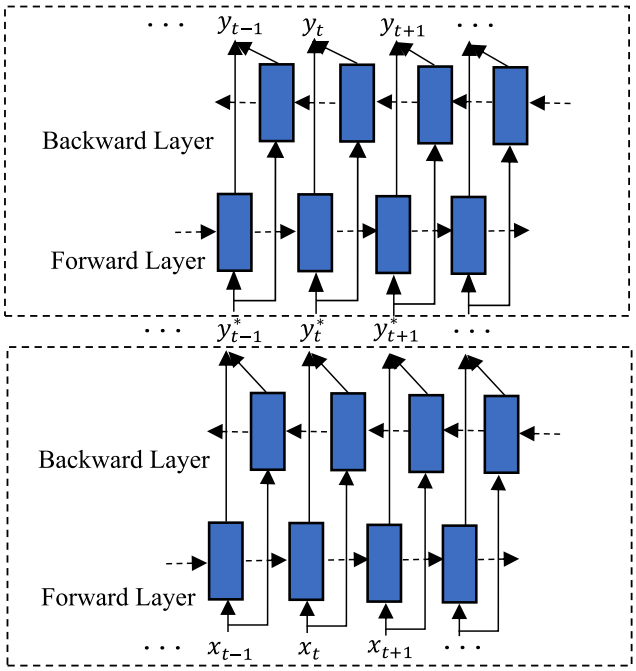

Fig. 5 Stacked Bi-LSTM network

can be estimated using an affinity function (scaler dot product, average) of the query and the corresponding key. Schematic structure of generic attention layer is presented in Figure 6. In our model, we use self-attention mechanism to generate opinion-sensitive context from the input-sequence. We use the outputs $Y = (y_1, y_2, \ldots y_t \ldots, y_n)$ of the final layer of stacked Bi-LSTM layer as query vector $Q^e \in R^{n \times D}$, key vector $K^e \in R^{n \times D}$ and value vector $V^e \in R^{n \times D}$. We use scaled dot product attention in our model since it is highly time and space efficient in practice. Equations 11 to 13 illustrate the calculation of attention distribution:

$$e^t = \frac{Q^e K^{eT}}{\sqrt{D}} \tag{11}$$

$$a^t = softmax(e^t) \tag{12}$$

$$Attention(Q, K, V) = a^t V^e \tag{13}$$

Instead of using single-head based attention distribution, we will use multi-head based self-attention [Figure 7] mechanism. The D dimensional query, key and value vectors are linearly projected to obtain transformed space query, key and value vectors of dimensions d= D/h each, where h is the number of parallel layers(heads) employed for calculating attention distribution. The projection parameters W_i^Q, W_i^K and $W_i^V \in R^{D \times d}$ are learned for optimal model performance. The h parallel layers attend to different semantic subspaces at different positions to contextually model information. The attention distributions obtained using each

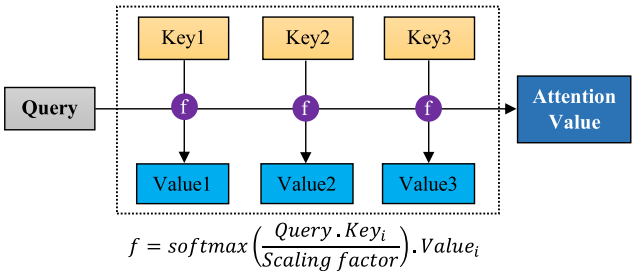

Fig. 6 Schematic Structure of Attention function

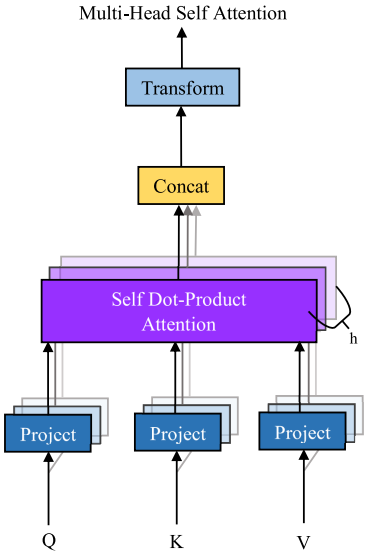

Fig. 7 Multi-Head Attention

of the parallel processing heads are concatenated and linearly projected to obtain the encoder context vector c^e. The involved calculations are illustrated in equations 14-15.

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{14}$$

$$c^e = MultiHead(Q,K,V) = Concat(head_1, \ldots head_h) \times W^o \tag{15}$$

where, W^o is a trainable parameter.

3.6 LSTM Based Decoder

FP2GN uses LSTM based decoder to calculate probabilistic vocabulary distribution for opinion summary generation. The decoder is inspired from pointer generator network [16] that efficiently handles the out of vocabulary (OOV) problem and hence,

gives efficient results with limited vocabulary. In the training phase, teacher forcing algorithm [54] is used.

– **Teacher Forcing Algorithm:** Models having recurrent connections from their outputs fed back into the next neuron (next timestep) may be trained using teacher forcing algorithm. Teacher forcing algorithm works by using the ground truth (expected) output from the training dataset y_t^* (at the current time step) as input into the next time step x_{t+1}. It facilitates quick and efficient training of the LSTM decoder network by incorporating the ground truth input summaries, $Y^* = \{y_i^* | i \in [1, n^{'}]\}$. At each decoding step, Teacher forcing algorithm minimizes the maximum likelihood loss (L_{tf}) as specified in equation 16.

$$L_{tf} = -\sum_{t=1}^{n^{'}} \log p(y_t^* | y_1^*, \ldots y_{t-1}^*, x) \tag{16}$$

where, x denotes the reference text.

But conventional teacher forcing algorithm uses actual(expected) summary words as input for the next time steps, rather than the generated summary words. Hence, the model is not trained to incorporate its own predictions for generating final summaries. However, ground truth summaries are not available in the testing phase. This discrepancy in model training leads to serious error accumulation and hence poor results. This train-test inconsistency arising from autoregressive generation models that use ground truth sequences at training time and predicted outputs at test time, is referred to as exposure bias. To deal with the issue of potential exposure bias, we introduce *teacher forcing ratio*.

– ***Teacher forcing Ratio:*** This study proposes the use of teacher forcing ratio α [Figure 8] as the probability of using the ground truth summary word (y_t^*) as input to the LSTM decoder [17]. Therefore, predicted output from the previous timestep $y_t^{'}$ is fed as input into the network with a probability of $(1-\alpha)$. α attempts to balance the skew in learned weights arising from the use of expected words for summary generation. Therefore, optimal selection of α can substantially counter the impact of exposure-bias on the performance of FP2GN. Skip-Gram Word2Vec method has been used for generating the word embeddings for pre-processed ground truth summary. This embedding layer is followed by LSTM layer.

The hidden state vector and cell state vector pertaining to the last layer of encoder, along with the start token embedding are used to initialize the decoder network. The following layers process the LSTM outputs for final summary generation:

– *Decoder Temporal Attention Layer:* The multi-head attention layer in encoder ensures that the requisite parts of input-sequence are adequately represented. But it does not solve the problem of repetition in the output summary, since decoder can generate repeated phrases owing to its own hidden states s_t. To solve this problem, information about the previously decoded output-sequence needs to be incorporated into the decoder [9]. Subsuming important thematic information from previous time stamps will lead to more structured predictions and repetition avoidance.

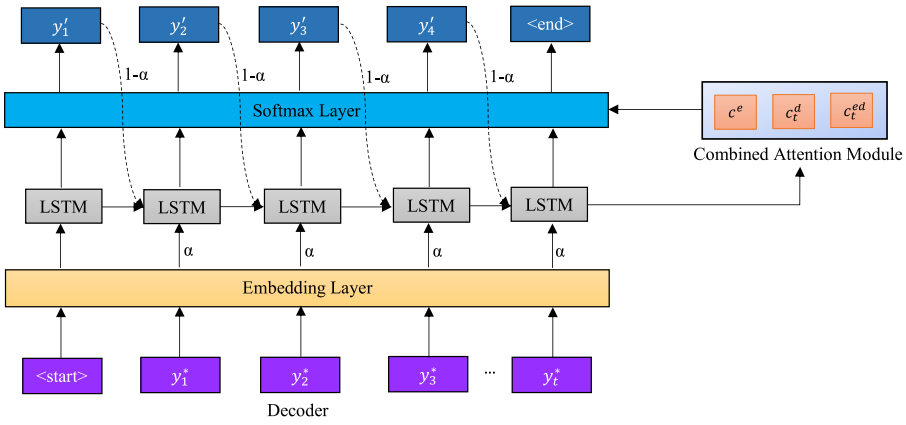

Fig. 8 Generic Workflow of LSTM based decoder

For each decoding timestep, temporal attention c_t^d is calculated using equations 17 to 19.

$$e_{tk}^d = s_t^{dT} W_{temp}^d s_k^d \tag{17}$$

$$a_{tk}^d = \frac{\exp(e_{tk}^d)}{\sum_{j=1}^{t-1} \exp(e_{tj}^d)} \tag{18}$$

$$c_t^d = \sum_{j=1}^{t-1} a_{tj}^d s_j^d \tag{19}$$

where, W_{temp}^d denotes trainable weight matrix and s_i^d indicates decoder hidden states.

Temporal Attention layer allows the decoder to attend to different positions in the decoded sequence up to the specified timestamp, and hence ensues repetition avoidance and context coherence.

– *Encoder-Decoder multi-head mutual attention Layer:* Multi head soft attention is used to calculate inter encoder-decoder context vector c_t^{ed}. Decoder hidden state (s_t) at decoder timestamp t is used as query vector Q^{ed} and encoder outputs $Y = (y_1, y_2, \ldots y_t \ldots, y_n)$ are used both as key K^{ed} and value V^{ed} vectors. The detailed calculations involved in evaluating encoder-decoder soft attention are shown in equations 20 to 22.

$$e_i^t = \tanh(W_y y_i + W_s s_t + b_{ed}) \tag{20}$$

$$a^t = softmax(e^t) \tag{21}$$

$$c_t^{ed} = \sum_i a_i^t y^i \tag{22}$$

where tanh denotes hyperbolic tangent function and W_y, W_s, b_{ed} are trainable parameters.

Inter encoder-decoder context assimilation using multi-head soft attention enhances the quality of aspect-oriented opinion summarization and stabilizes the training process. It enables adequate representation of the input sequence at each decoding step to maintain the consistency of factual and sentiment-oriented details.

– *Softmax layer:* Encoder context vector c^e, decoder temporal context vector c_t^d, encoder-decoder context vector c_t^{ed} and decoder hidden state s_t at decoder timestamp t are processed in the softmax layer to obtain final vocabulary distribution. As discussed before, we use a pointer generator network to calculate the probabilistic distribution for words present in the vocabulary. Pointer generator network [16] handles the OOV (out of vocabulary) problem of most neural seq2seq models by allowing the decoder to both, copy words from input sequence with pointing probability $p_{point} \in [0,1]$ and generating new words with generation probability $p_{gen} \in [0,1]$. Hence, our model uses pointer generator inspired LSTM based decoder for aspect-oriented opinion summarization. The generation probability p_{gen} acts as a soft switch for choosing between generating a word from vocabulary and copying a word from input review through pointing. The generation probability p_{gen} is calculated using equation 23.

$$p_{gen} = \tanh(w_e^T c^e + w_{ed}^T c_t^{ed} + w_d^T c_t^d + w_s^T s_t + w_x^T x_t + b_{pgn}) \tag{23}$$

where $w_e^T, w_{ed}^T, w_d^T, w_s^T, w_x^T$ and b_{pgn} are learnable parameters and x_t is the LSTM input.

The extended document vocabulary is defined as the union of base vocabulary and all the words present in the source reviews. Equations 24-25 illustrate the evaluation of probabilistic vocabulary distribution over the extended vocabulary.

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} c_{t_i}^{ed} \tag{24}$$

$$P_{vocab} = softmax(V^{'}(V[s_t, c^e, c_t^{ed}, c_t^d] + b) + b^{'}) \tag{25}$$

where $V^{'}, V, b^{'}$ and b are learnable parameters.

For optimum performance of the model FP2GN, maximum likelihood loss associated with teacher forcing algorithm (equation 16), and the cross-entropy loss (equation 26) must be minimized. We hence define a multi-objective loss function that linearly combines cross entropy loss (coefficient ω) and teacher forcing maximum likelihood loss (coefficient $1-\omega$) to yield context-coherent, meaningful, and consistent summaries. The multi-objective loss (equation 27) ensures that the exposure bias is minimized, and human readability is better captured in the decoded output sequence. Equations 26-28 illustrate the loss functions.

$$\text{cross entropy loss}_t = -\log P(w_t^*) \tag{26}$$

$$L_{mo}^t = \omega.(-\log P(w_t^*)) + (1-\omega).L_{tf} \tag{27}$$

$$loss = \frac{1}{T} \sum_{t=0}^{T} L_{mo}^t \tag{28}$$

where,
w_t^* depicts the decoded output at time t
T denotes the total time-steps

4 Dataset Examples

In this section, we will discuss a few dataset examples to aid better understanding of the workflow of the proposed FP2GN model.

Example 1: Positive opinion polarity

Review: 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.

Step i: Data Pre-processing

['buy', 'vitality', 'canned', 'dog', 'food', 'product', 'find', 'good', 'quality', 'product', 'look', 'like', 'stew', 'process', 'meat', 'smell', 'better', 'labrador', 'finicky', 'appreciate', 'product', 'better']

Step ii: Opinion Feature Mapping

['buy'/O, 'vitality'/O, 'canned'/O, 'dog'/O, 'food'/B, 'product'/O, 'find'/O, 'good'/O, 'quality'/B, 'product'/O, 'look'/O, 'like'/O, 'stew'/O, 'process'/O, 'meat'/B, 'smell'/O, 'better'/O, 'labrador'/O, 'finicky'/O, 'appreciate'/O, 'product'/O, 'better'/O]

Step iii: Summary generation

Reference Summary: good quality dog food

Generated Summary: good quality food

Example 2: Negative opinion polarity

Review: I fed this to my Golden Retriever, and he hated it. He wouldn't eat it, and when he did, it gave him terrible diarrhoea. We will not be buying this again. It's also super expensive.

Step i: Data Pre-processing

['feed', 'golden', 'retriever', 'hate', 'eat', 'give', 'terrible', 'diarrhoea', 'buy', 'super', 'expensive']

Step ii: Opinion Feature Mapping

['feed'/O, 'golden'/O, 'retriever'/O, 'hate'/O, 'eat'/O, 'give'/O, 'terrible'/O, 'diarrhoea'/O, 'buy'/O, 'super'/O, 'expensive'/O]

Step iii: Summary generation

Reference Summary: bad product not worth buy

Generated Summary: bad not buy

Example 3: Mixed opinion polarity

Review: Popcorn has great colour but taste is average and due to shipping costing more than product, I would not buy again.

Step i: Data Pre-processing

['popcorn', 'great', 'colour', 'taste', 'average', 'shipping', 'cost', 'product', 'buy']

Step ii: Opinion Feature Mapping

['popcorn'/B, 'great'/O, 'colour'/B, 'taste'/B, 'average'/O, 'shipping'/O, 'cost'/O, 'product'/O, 'buy'/O]

Step iii: Summary generation

Reference Summary: good colour average taste

Generated Summary: good colour average taste

These examples verify that FP2GN model gives non-repetitive, context adhering and factually consistent summaries that retain the core aspect-based sentiment of reference-text.

5 Experimentation and Result

Amazon fine foods [48] dataset is used to evaluate the performance of the proposed model FP2GN. Since the dataset is too large to be processed on our local system, a total of 10000 reviews are randomly chosen for experimentation purposes. K-fold cross validation method is used for validation and testing. In each iteration, 6000 reviews are used in the training phase, and 2000 reviews each in the validation and the testing phases.

Figures 9 and 10 show the sentiment distribution across the 10000 reviews and the associated generated summaries, respectively. Both the curves illustrate similar sentiment score distribution validating the premise that the generated summaries have high opinion coverage.

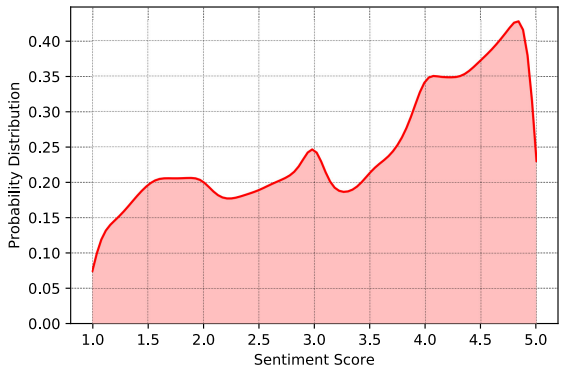

Fig. 9 Sentiment distribution across reviews

The rest of this section is organised as follows: section 5.1 elucidates the hyperparameter setting for optimal performance of FP2GN, section 5.2 illustrates the

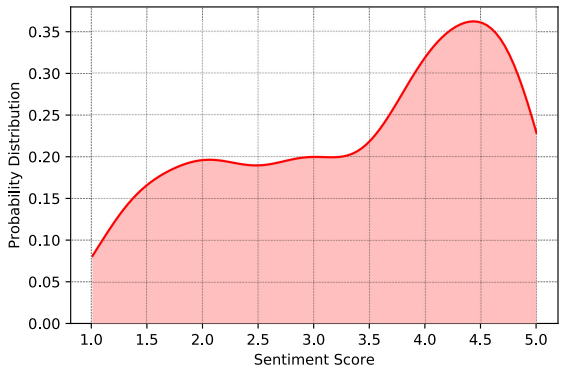

Fig. 10 Sentiment distribution across review summaries

performance of our proposed model. Section 5.3 compares the results of the study with competing opinion summarizers.

5.1 Hyperparameter tuning

Model parameters must be optimally selected to ensure superlative performance. Validation data has been used to fine-tune model parameters so as to achieve optimal results. Hyperparameters and their corresponding values used in this work are enlisted in table 6. Figure 11 shows the variation of validation loss with respect to teacher forcing threshold α. It can be observed that best results are achieved with $\alpha = 0.4$.

Table 6 Hyperparameter tuning for FP2GN

Hyperparameter	**Value**
Dimension of Skip-Gram Embeddings	256
Mini-batch size	50
Latent dimension of LSTM	300
Optimiser	Adagrad
Learning rate	0.15
Regularization	Dropout Operation
Dropout rate	0.08: Word embeddings; 0.2: Bi-LSTM
Teacher forcing threshold (α)	0.4
Context similarity threshold (δ)	0.4
Multi-objective loss coefficient (ω)	0.8

The weight matrices are randomly initialized using standard orthogonal distribution with seeding value of 2. All the bias vectors are initialized to zero matrices except the forget gate bias, that is initialized to unity matrix.

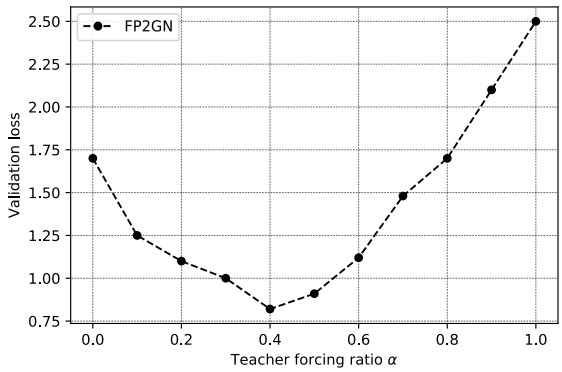

Fig. 11 Variation of validation loss with respect to α

5.2 Performance Results

The performance of the proposed deep learning model has been evaluated using three performance evaluation metrics: ROUGE-1, ROUGE-2 and ROUGE-L. The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score is a measure of the consistency between n-gram occurrences in the reference and the generated summaries. A high ROUGE value is indicative of high-quality summary whereas, a low ROUGE value indicates that the generated summary is not good enough to contextually represent the original text. Table 7 enlists the evaluation metrics used in the study.

Table 7 Evaluation metrics used to evaluate the model

Evaluation Metric	**Description**
ROUGE-1	Overlap of unigrams between reference and generated summaries.
ROUGE-2	Overlap of bigrams between reference and generated summaries.
ROUGE-L	Longest common subsequence as a measure of overlap between the reference and generated summaries.

Table 8 illustrates the ROUGE values achieved by our model. The high ROUGE values indicate that the generated opinion summaries are of high quality and can adequately represent the sentiment of the input review. The results validate the efficacy of the proposed opinion summarization technique.

Figure 12 shows the variation of the multi-objective training and validation losses. Both the training and validation losses exponentially fall as the training proceeds. We continue the training process for 1000 iterations where the multi-objective losses for both the phases become almost equal and successive iterations do not bring about significant improvement in model performance.

Table 8 Performance of our model

Evaluation Metric	Value
ROUGE-1	86.04
ROUGE-2	70.12
ROUGE-L	88.51

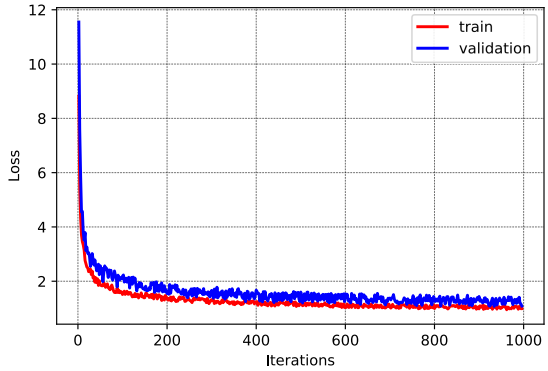

Fig. 12 Training-Validation Loss curve

We compare the ROUGE performance of single-head encoder attention-based feature pooled pointer generator network with multi-head self-attention based FP2GN, and the results are demonstrated in Figure 13. It is evident that multi-head self-attention outperforms its single-head counterpart. Additionally, multi-head encoder attention stabilizes the training process and ensures adequate representation of different semantic-subspaces in encoded context-vector.

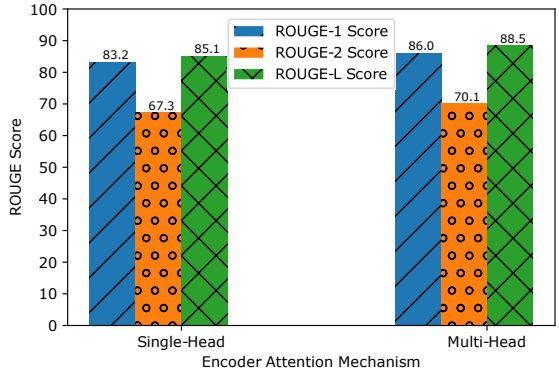

Fig. 13 Comparative Analysis of Encoder attention mechanisms

Temporal attention ensures that the input-sequence is duly represented at each decoding step. Hence, FP2GN selects temporal decoder attention over coverage mechanism [16] for repetition avoidance. The comparison of the two approaches is illustrated in Figure 14.

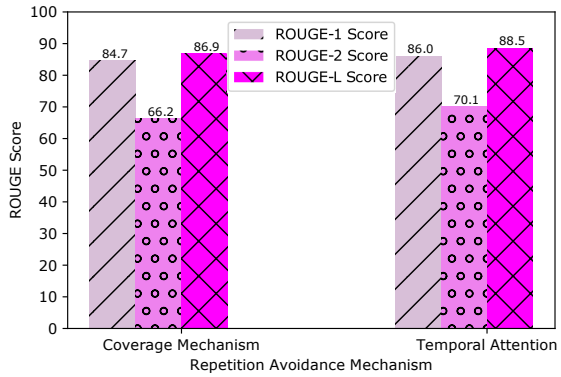

Fig. 14 Comparative Analysis of Repetition Avoidance Mechanisms

In this study, we propose the use of CNN-like weighted average pooling technique for aspect-fused context representation. We compare the use of max-pooling, average pooling, opinion-word target concatenation (OWTC), opinion-word target average pooling (OWTAP), two-step sentence attention based LSTM (TSA-LSTM) [4] and target concatenation (TC) as potential aspect-fusion techniques [55]. Max-pooling picks the higher magnitude embedding dimension between the opinion feature and other words. OWTC concatenates the nearest opinion word (eg. good, great) embedding with that of opinion feature present in a phrase for aspect-encapsulation. OWTAP average pools the opinion word and opinion target embeddings and retains all other embeddings in their original form. TSA-LSTM uses LSTM for encoding review sentences and attends to the opinion features and sentences in a hierarchical fashion. TC concatenates all positional embeddings with that of opinion feature. The performance of TC is compatible with average pooling technique, but it is not time and space efficient. The trainable parameters in TC are exponentially more than the proposed average pooling technique. Hence, TC is not a recommended aspect-fusion method. FP2GN outperforms TSA-LSTM since stacked Bi-LSTM can effectively model past and future contexts, and multi-head attention selectively attends to aspect-fused context embeddings. Also, weighted average pooling technique embeds the characteristics of opinion feature into all the context words to aid adequate representation of the aspect-information and neighborhood semantics at each decoding step. The comparison of various aspect-fusion techniques is shown in Figure 15.

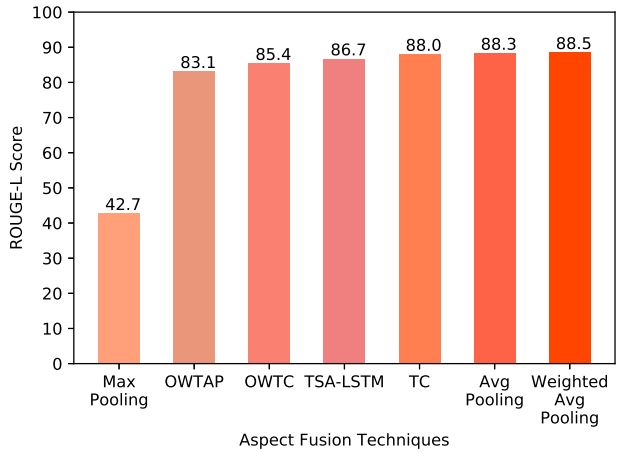

Fig. 15 Comparative analysis of various Aspect fusion techniques

5.3 Comparison with baselines

We compare the proposed model with various baseline approaches as well as with the methodologies proposed in other state of the art research works. Table 9 compares FP2GN with the baseline Abstractive model ABS proposed by Nallapati et al. [33], attention-based sequence to sequence encoder decoder model [16] and extractive summarizer SummaRuNNer [28]. The results validate that the use of multi-head, temporal and mutual attention certainly improves the performance of deep neural opinion summarizers.

Table 9 Comparison of FP2GN with baseline models

Model	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
ABS	73.18	49.37	77.31
Seq-to-seq + attn	64.86	43.84	68.27
SummaRuNNer	80.49	62.71	81.63
FP2GN	86.04	70.12	88.51

Table 10 compares the ROUGE-1, ROUGE-2 and ROUGE-L values achieved by FP2GN with those obtained using the state-of-the-art methodologies proposed by other researchers. ASDKGA [27] does not achieve appreciable opinion summarization performance because it cannot ensure repetition avoidance and is highly dependent on the factual consistency of domain-knowledge. PGC [16] performs better than ABS [33] because it can effectively handle the out of vocabulary problem. PGC also uses the coverage mechanism to eliminate repetitions in generated summary. DeepRL [37] and GANsum [38] outperform PGC because they use reinforcement policy learning. MARS [9] uses text categorization and multi-factor attention for efficient aspect-based opinion summarization. FP2GN performs even better than MARS

(with a ROUGE-L gain of approximately 2%) owing to the use of opinion feature extraction, feature pooling, temporal and mutual attention mechanism. Table 10 and Figure 16 substantiate the efficacy of the proposed model for aspect-based opinion summarization.

Repetition avoidance and robustness to OOV words are important characteristics of opinion summarization. Simultaneously, state-of-the-art opinion summarizers are expected to be independent of the categorical specifications (domain) of the training data. It has also been observed that techniques like policy learning (reinforcement learning), temporal attention and aspect-information incorporation (embedding context-information into opinion summarizers) boost the performance of abstractive opinion summarizers. Therefore, FP2GN is qualitatively compared to the baseline models with respect to these performance characteristics and results are summarized in Table 11.

Table 10 Comparison of FP2GN with state-of-the-art research methodologies

Model	ROUGE-1	ROUGE-2	ROUGE-L
ASDKGA	67.51	52.91	68.39
PGC	81.84	64.15	83.18
DeepRL	82.12	65.09	84.31
GANsum	82.64	66.12	84.31
MARS	84.13	68.28	86.15
FP2GN	86.04	70.12	88.51

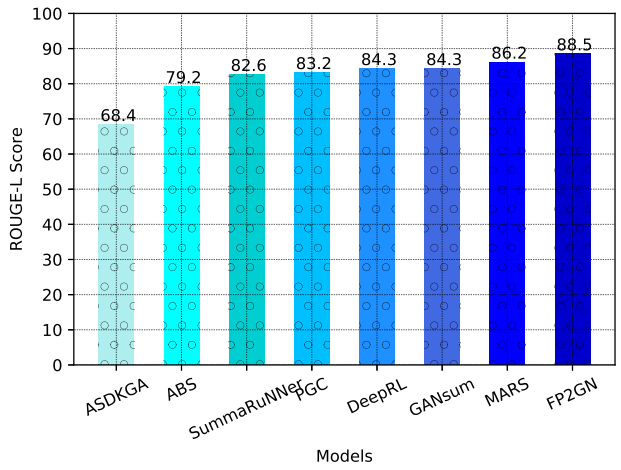

Fig. 16 Comparison of FP2GN with state-of-the-art research methodologies

Table 11 Qualitative comparison of baseline models with FP2GN

Models	Criteria					
	Repetion Avoidance	**OOV Words**	**Dependence on Domain Knowledge**	**RL Policy Learning**	**Temporal Attention**	**Aspect Incorporation**
PGC	Yes	Yes	No	No	No	No
DeepRL	Yes	Yes	No	Yes	Yes	No
GANSum	Yes	Yes	No	Yes	No	No
ABS	Yes	Yes	No	No	Yes	Yes
Seq-Seq + Attn	No	No	No	No	No	No
MARS	Yes	Yes	No	Yes	No	Yes
ASKDGA	No	Yes	Yes	No	No	Yes
FP2GN	Yes	Yes	No	No	Yes	Yes

5.4 Applications

The proposed model FP2GN can be easily integrated into real-life applications. The generated opinion summaries can be used to keep a tab on online user sentiment, which can be manipulated for several predictive analytics tasks. For instance, politicians can use Twitter opinions to predict the likelihood of election outcomes. Opinion summarization also finds application in domains like Customer Relationship Management, Business Intelligence and Social Media Analytics. In this study, FP2GN has been used to collect, integrate, and present opinionated data in a concise form. The opinion summaries can be analysed using modern day Business Intelligence tools to maximize customer satisfaction. The aspect-based features extracted by the model can be used as KPIs (key performance indicators) to identify trends and glean intelligence about customer behaviour and business operations. The insights provided by opinion summaries can facilitate business planning, product characterization and strategic marketing. Hence, FP2GN is a time, space and effort-efficient technique to extract sentiment-features and generate opinion summaries, that can be further exploited to gain business advantage.

6 Conclusion

This study proposes a sentic computing based opinion summarizer FP2GN that successfully counteracts the problems of exposure bias, repetition, generalization and factual inconsistency in generated summaries. The model harnesses the success of pointer generator networks and augments it with self-attention, temporal-attention and mutual soft-attention mechanisms to integrate sentiment information into the output summaries. Additionally, FP2GN leverages sentic computing techniques for opinion feature identification and target-fused thematic review representation. Amazon fine foods dataset has been used to validate the results. The model outperformed baselines with ROUGE-1 score of 86.04%, ROUGE-2 score of 70.12% and ROUGE-L score of 88.51%. The remarkable performance of FP2GN indicates that target encapsulation and multi-head attention mechanism can enhance baseline sequence-

sequence models to generate specialised human-readable summaries. The work has opened avenues for the use of CNN-like pooling techniques to capture the neighborhood semantic properties of text for aspect-oriented sentiment analysis tasks. As a possible future direction, actor-critic reinforcement learning policy can be used for improving the readability of summaries. We also plan to use SenticNet 6 [56] to improve the performance of opinion feature extraction in FP2GN. Besides, more sophisticated n-gram based techniques can be devised for opinion feature extraction and target-fused context representation.

Funding Information The author(s) received no financial support for the research, authorship, and/or publication of this article.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. R. Satapathy, A. Singh, and E. Cambria, "Phonsenticnet: A cognitive approach to microtext normalization for concept-level sentiment analysis," in *International Conference on Computational Data and Social Networks*, pp. 177–188, Springer, 2019.
2. E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment analysis is a big suitcase," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 74–80, 2017.
3. N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, and E. Cambria, "Dialoguernn: An attentive rnn for emotion detection in conversations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6818–6825, 2019.
4. Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, "Sentic lstm: a hybrid network for targeted aspect-based sentiment analysis," *Cognitive Computation*, vol. 10, no. 4, pp. 639–650, 2018.
5. A. Picasso, S. Merello, Y. Ma, L. Oneto, and E. Cambria, "Technical analysis and sentiment embeddings for market trend prediction," *Expert Systems with Applications*, vol. 135, pp. 60–70, 2019.
6. D. A. Economics, "Economic contribution of the great barrier reef," 2013.
7. L. Koesten, E. Simperl, T. Blount, E. Kacprzak, and J. Tennison, "Everything you always wanted to know about a dataset: studies in data summarisation," *International Journal of Human-Computer Studies*, vol. 135, p. 102367, 2020.
8. D. K. Gaikwad and C. N. Mahender, "A review paper on text summarization," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 3, pp. 154–160, 2016.
9. M. Yang, Q. Qu, Y. Shen, Q. Liu, W. Zhao, and J. Zhu, "Aspect and sentiment aware abstractive review summarization," in *Proceedings of the 27th international conference on computational linguistics*, pp. 1110–1120, 2018.
10. M. E. Moussa, E. H. Mohamed, and M. H. Haggag, "A survey on opinion summarization techniques for social media," *Future Computing and Informatics Journal*, vol. 3, no. 1, pp. 82–109, 2018.
11. E. Cambria, M. Grassi, A. Hussain, and C. Havasi, "Sentic computing for social media marketing," *Multimedia tools and applications*, vol. 59, no. 2, pp. 557–577, 2012.
12. E. Cambria, T. Benson, C. Eckl, and A. Hussain, "Sentic proms: Application of sentic computing to the development of a novel unified framework for measuring health-care quality," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10533–10543, 2012.

13. E. Cambria, A. Hussain, C. Havasi, and C. Eckl, "Sentic computing: Exploitation of common sense for the development of emotion-sensitive systems," in *Development of Multimodal Interfaces: Active Listening and Synchrony*, pp. 148–156, Springer, 2010.
14. E. Cambria, J. Fu, F. Bisio, and S. Poria, "Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis.," in *AAAI*, pp. 508–514, 2015.
15. E. Cambria, A. Hussain, C. Havasi, and C. Eckl, "Senticspace: visualizing opinions and sentiments in a multi-dimensional vector space," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 385–393, Springer, 2010.
16. A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.
17. J. Li, C. Zhang, X. Chen, Y. Cao, P. Liao, and P. Zhang, "Abstractive text summarization with multi-head attention," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.
18. R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, 2004.
19. S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," *Text mining: applications and theory*, vol. 1, pp. 1–20, 2010.
20. E. Cambria, A. Hussain, T. Durrani, C. Havasi, C. Eckl, and J. Munro, "Sentic computing for patient centered applications," in *IEEE 10th International Conference on Signal Processing Proceedings*, pp. 1279–1282, IEEE, 2010.
21. E. Cambria, S. Poria, D. Hazarika, and K. Kwok, "Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 1795–1802, 2018.
22. S.-A. Bahrainian and A. Dengel, "Sentiment analysis and summarization of twitter data," in *2013 IEEE 16th International Conference on Computational Science and Engineering*, pp. 227–234, IEEE, 2013.
23. W. Luo, F. Zhuang, Q. He, and Z. Shi, "Exploiting relevance, coverage, and novelty for query-focused multi-document summarization," *Knowledge-Based Systems*, vol. 46, pp. 33–42, 2013.
24. N. Pavlopoulou and E. Curry, "Using embeddings for dynamic diverse summarisation in heterogeneous graph streams," in *2019 First International Conference on Graph Computing (GC)*, pp. 5–12, 2019.
25. J. Guo, Y. Lu, T. Mori, and C. Blake, "Expert-guided contrastive opinion summarization for controversial issues," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1105–1110, 2015.
26. M. Subramaniam and V. Dalal, "Test model for rich semantic graph representation for hindi text using abstractive method," *International Research Journal of Engineering and Technology (IRJET)*, vol. 2, no. 2, 2015.
27. Q. A. Al-Radaideh and D. Q. Bataineh, "A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms," *Cognitive Computation*, vol. 10, no. 4, pp. 651–669, 2018.
28. R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
29. J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "A decomposition-based multi-objective optimization approach for extractive multi-document text summarization," *Applied Soft Computing*, p. 106231, 2020.
30. M. Mohd, R. Jan, and M. Shah, "Text document summarization using word embedding," *Expert Systems with Applications*, vol. 143, p. 112958, 2020.
31. M. Rajangam and C. Annamalai, "Extractive document summarization using an adaptive, knowledge based cognitive model," *Cognitive Systems Research*, vol. 56, pp. 56–71, 2019.
32. T. Hirao, M. Nishino, Y. Yoshida, J. Suzuki, N. Yasuda, and M. Nagata, "Summarizing a document by trimming the discourse tree," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 2081–2092, 2015.
33. R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, *et al.*, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv preprint arXiv:1602.06023*, 2016.
34. J. Zheng, Z. Zhao, Z. Song, M. Yang, J. Xiao, and X. Yan, "Abstractive meeting summarization by hierarchical adaptive segmental network learning with multiple revising steps," *Neurocomputing*, vol. 378, pp. 179–188, 2020.

35. D. S. Moirangthem and M. Lee, "Abstractive summarization of long texts by representing multiple compositionalities with temporal hierarchical pointer generator network," *Neural Networks*, vol. 124, pp. 1–11, 2020.
36. R. Adelia, S. Suyanto, and U. N. Wisesty, "Indonesian abstractive text summarization using bidirectional gated recurrent unit," *Procedia Computer Science*, vol. 157, pp. 581–588, 2019.
37. R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *arXiv preprint arXiv:1705.04304*, 2017.
38. L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative adversarial network for abstractive text summarization," in *Thirty-second AAAI conference on artificial intelligence*, 2018.
39. M. Dragoni, M. Federici, and A. Rexha, "Reus: a real-time unsupervised system for monitoring opinion streams," *Cognitive Computation*, vol. 11, no. 4, pp. 469–488, 2019.
40. S. Gupta and P. Halder, "A hybrid lexicon-based sentiment and behaviour prediction system," in *Advances in Control, Signal Processing and Energy Systems*, pp. 67–77, Springer, 2020.
41. F. Bisio, C. Meda, P. Gastaldo, R. Zunino, and E. Cambria, "Concept-level sentiment analysis with senticnet," in *A Practical Guide to Sentiment Analysis*, pp. 173–188, Springer, 2017.
42. F. Z. Xing, E. Cambria, and R. E. Welsch, "Natural language based financial forecasting: a survey," *Artificial Intelligence Review*, vol. 50, no. 1, pp. 49–73, 2018.
43. Z. Deng, F. Ma, R. Lan, W. Huang, and X. Luo, "A two-stage chinese text summarization algorithm using keyword information and adversarial learning," *Neurocomputing*, 2020.
44. H.-X. Pan, H. Liu, and Y. Tang, "A sequence-to-sequence text summarization model with topic based attention mechanism," in *International Conference on Web Information Systems and Applications*, pp. 285–297, Springer, 2019.
45. P. Wu, X. Li, S. Shen, and D. He, "Social media opinion summarization using emotion cognition and convolutional neural networks," *International Journal of Information Management*, vol. 51, p. 101978, 2020.
46. Q. Zhou, R. Xia, and C. Zhang, "Online shopping behavior study based on multi-granularity opinion mining: China versus america," *Cognitive Computation*, vol. 8, no. 4, pp. 587–602, 2016.
47. A. Abdi, S. M. Shamsuddin, and R. M. Aliguliyev, "Qmos: Query-based multi-documents opinion-oriented summarization," *Information Processing & Management*, vol. 54, no. 2, pp. 318–338, 2018.
48. J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 897–908, 2013.
49. S. Hu, A. Kumar, F. Al-Turjman, S. Gupta, S. Seth, *et al.*, "Reviewer credibility and sentiment analysis based user profile modelling for online product recommendation," *IEEE Access*, vol. 8, pp. 26172–26189, 2020.
50. L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Third Workshop on Very Large Corpora*, 1995.
51. E. Loper and S. Bird, "Nltk: the natural language toolkit," *arXiv preprint cs/0205028*, 2002.
52. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
53. D. Wang and E. Nyberg, "A long short-term memory model for answer sentence selection in question answering," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 707–712, 2015.
54. R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.
55. Z. Fan, Z. Wu, X. Dai, S. Huang, and J. Chen, "Target-oriented opinion words extraction with target-fused neural sequence labeling," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2509–2518, 2019.
56. E. Cambria, Y. Li, F. Z. Xing, S. Poria, and K. Kwok, "Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis," *CIKM'20, Oct 20-24*, 2020.