ConsistSum: Unsupervised Opinion Summarization with the Consistency of Aspect, Sentiment and Semantic

Wenjun Ke[1,2], Jinhua Gao*[1], Huawei Shen[1], Xueqi Cheng[1]
1. Data Intelligence System Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
2. University of Chinese Academy of Sciences, Beijing, China
kewenjun2191@163.com,{gaojinhua,shenhuawei,cxq}@ict.ac.cn

ABSTRACT

Unsupervised opinion summarization techniques are designed to condense the review data and summarize informative and salient opinions in the absence of golden references. Existing dominant methods generally follow a two-stage framework: first creating the synthetic "review-summary" paired datasets and then feeding them into the generative summary model for supervised training. However, these methods mainly focus on semantic similarity in synthetic dataset creation, ignoring the consistency of aspects and sentiments in synthetic pairs. Such inconsistency also brings a gap to the training and inference of the summarization model.

To alleviate this problem, we propose ConsistSum, an unsupervised opinion summarization method devoting to capture the consistency of aspects and sentiment between reviews and summaries. Specifically, ConsistSum first extracts the preliminary "review-summary" pairs from the raw corpus by evaluating the distance of aspect distribution and sentiment distribution. Then, we refine the preliminary summary with the constrained Metropolis-Hastings sampling to produce highly consistent synthetic datasets. In the summarization phase, we adopt the generative model T5 as the summarization model. T5 is fine-tuned for the opinion summarization task by incorporating the loss of predicting aspect and opinion distribution. Experimental results on two benchmark datasets, *i.e.*, Yelp and Amazon, demonstrate the superior performance of ConsistSum over the state-of-the-art baselines.

CCS CONCEPTS

• **Information systems → Information systems applications**.

KEYWORDS

opinion summarization, unsupervised method, consistency enhancement

ACM Reference Format:
Wenjun Ke[1,2], Jinhua Gao*[1], Huawei Shen[1], Xueqi Cheng[1]. 2022. ConsistSum: Unsupervised Opinion Summarization with the Consistency of Aspect, Sentiment and Semantic. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498463

*Corresponding author.

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498463

1 INTRODUCTION

Social media generates a large amount of review data on the Internet that varies among topics, *e.g.*, online goods [34], government policies [25], and health care [16]. To facilitate the decision-making process, many opinion summarization techniques have been proposed to filter valuable opinions out of the massive review data [1, 2, 6, 7, 9, 18, 24, 41, 42, 44].

Opinion summarization techniques can be categorized into two families, *i.e.*, supervised and unsupervised, based on their working nature. However, gold reference summaries are usually insufficient in various domains, limiting the applicability of supervised methods and making the unsupervised methods more practical. Previous unsupervised studies typically fall into two paradigms: extracting salient sentences from original reviews directly [4, 22, 31, 36, 46], or abstracting opinion summaries by condensing the meaning of reviews [1, 2, 7, 9, 41, 43]. Extractive approaches cluster features of review segments, and calculate the salience and redundancy of each segment to extract summaries that satisfy the maximum length limit greedily. However, the extractive method might raise issues such as loss of important information and lack of language fluency. In comparison with extractive methods, abstractive methods utilize auto-encoder [7, 9] or sequence-to-sequence model [1, 41, 43] to capture the features of reviews as a whole and generate more robust summaries. For example, in Figure 1, the abstractive summary is more coherent and comprehensive than the extractive one. To resolve the lack of labeled data, the abstractive approach mainly focuses on creating synthetic "review-summary" datasets. It first selects a pre-existing review sentence and treats it as a pseudo-summary $\tilde{y}$. Then, various greedy operations, such as noising the pseudo-summary [2] or content planning scheme [1], have been proposed to align the reviews x corresponding to $\tilde{y}$. Finally, the generated synthetic datasets are utilized for training the summarization model in a supervised way.

The key to the success of unsupervised abstractive methods lies in constructing high-quality synthetic datasets. Existing methods produce reviews x close to the pseudo-summary $\tilde{y}$ based on their semantic similarity. However, such a strategy brings two disadvantages. On the one hand, the synthetic datasets mainly focus on the semantic correlation between reviews and summaries, ignoring the significance of aspect and sentiment consistency with respect to the opinion summarization task. This inconsistency might bring a

gap to the training and inference of the summarization model. On the other hand, the greedy strategy often concentrates the reviews on specific topics in synthetic datasets, while realistic datasets often cover diverse topics. Such difference further downgrades the performance of existing methods.

Review 1: I did however really like the wontons in red chili sauce.
Review 2: Although I am not a big fan of beef noodles, I still wanted to come here and try their other dish. I ordered pig feet, noodle soup. … I think the beef noodle soup would've been more filling. But this place seems to be very popular as people kept coming in.
Review 3: The restaurant is really tiny and more of a cafe. The beef stew noodle is so perfect. Not too salty. Just enough beef, bok choy, and handmade noodles to satisfy any appetite. It is a milder soup but the saltiness of the pork chop helps to balance it.
Review 4: Zero star ... They have really bad service ... wrong order, bad attitude, even worse, no apology for their mistake .... and their food are horrible x 5 ... I will never ever come here again.
Review n: …

Extractive summary: The beef stew noodle is so perfect. Not too salty. The restaurant is really tiny and more of a cafe. They have really bad service, wrong order, bad attitude, and their food are horrible. I will never ever come here again.

Abstractive summary: This is a decent place to go for chinese food. I really like the beef noodle soup. It has the perfect amount of salt. I wish the place wasn't so small and stuffy. I just didn't feel comfortable at all. Next time I'll just place a to go order .

Figure 1: An illustration of reviews and summaries of a restaurant, in which colors represent different aspects.

To alleviate these issues, we propose ConsistSum, an unsupervised method that aims to maintain the consistency between reviews and summaries in terms of aspect, sentiment, and semantic for opinion summarization. Specifically, we first design the consistency distance metric to measure the consistency of aspect distribution and sentiment distribution to extract the preliminary "review-summary" datasets. This process can enhance the diversity of selected reviews while guaranteeing the consistency of preliminary datasets. Then, we further refine the preliminary pseudo-summary to improve its expressiveness. Motivated by the success of constrained sentence generation [8, 10, 27, 28, 30, 45], we adapt the Metropolis-Hastings sampling method (CGMH) [30, 38] to the BERT [19] model to edit the preliminary pseudo-summary progressively, conditioning on the consistency distance between the reviews and the preliminary summary. This step can further shorten the distance between reviews and corresponding pseudo-summaries and produce highly consistent synthetic datasets. Finally, we adopt the generative model T5 [35] as the summarization model. To better adapt T5 to the opinion summarization task, we train T5 in the multi-task learning schema that incorporates the loss of predicting aspect and sentiment distribution.

Our major contributions can be summarized as follows:

- We propose a novel consistency distance metric to pair the "review-summary" samples that possess more consistency in aspect and sentiment distribution.
- We adapt the constrained generation model to refining the preliminary pseudo-summary of synthetic datasets, further enhancing its expressiveness and shortening its consistency distance to the reviews.
- Combining the above two modules, we propose a novel unsupervised model, ConsistSum, to generate more coherent and comprehensive opinion summaries.
- Experimental results on Yelp and Amazon datasets demonstrate that ConsistSum can outperform state-of-the-art methods. Further analysis shows that the generated summaries of ConsistSum do keep higher consistency.

2 RELATED WORK

In this section, we introduce the related work under two mainstream branches: extractive approaches and abstractive approaches.

2.1 Extractive Opinion Summarization

The general paradigm of extractive summarization is to cluster the review segments, and iteratively extract the center of segments for splicing to get the summary [14]. Erkan *et al.* [11]used a page-rank-like algorithm based on word frequency to produce the review centroid and select review segments. Nallapati *et al.* [31] argued that the selected sentences should possess high salience, informativeness and novelty. Rossiello *et al.* [36] proposed that the semantic feature (*e.g.*, the word vector), rather than word frequency, can be appropriate to flatten the representation of synonyms in clustering. Angelidis *et al.* [4] proposed the MATE model, which considered the salience of aspect correlation when ranking review segments, and produced summaries satisfying the maximum limited length with a greedy strategy. Zhao *et al.* [46] and Lee *et al.* [22] conducted the integer linear programming algorithm (ILP) for global salience optimization, achieving more comprehensive results than that with greedy strategies.

Although extractive methods are efficient and easy to explain, they would suffer from some issues such as key information loss and utterance incoherence [1].

2.2 Abstractive Opinion Summarization

Abstractive methods devote to understand reviews from a holistic perspective to produce a concise and coherence summary covering the original core information. Previous works have been studied in three scenarios: supervised [3], semi supervised [6, 41] and unsupervised [1, 2, 7, 9]. In fact, the golden reference summaries are lacking in most cases. Thus, studies in an unsupervised situation would be more meaningful. Specifically, Chu *et al.* [9] leveraged the auto-encoder model to retrieve specific representation of each review and produce the summary. Bražinskas *et al.* [7] applies the variation autoencoder (VAE) [12] instead of vanilla auto-encoder to facilitate the correlation between summary and reviews and achieves better performance. Such design essentially enhances the model's self-awareness towards samples using reconstruction loss. Moreover, creating "review-summary" paired datasets for the supervised training could be a more effective method. Amplayo *et al.* [1, 2] conducted the noising strategy or the content planning induction to extract the representative review as the pseudo-summary, and reverse to align corresponding reviews, converting the unsupervised scenario to supervised scenario and training a generation model for summarization.

Existing abstractive methods have achieved promising performance. However, they focus on semantic similarity between summary and reviews, ignoring the importance of aspect and sentiment and the diversity of synthetic datasets.

3 METHOD

Given the raw corpus with n reviews $x_{raw} = \{x_1, x_2, \ldots, x_n\}$ with respect to a specific object T (*e.g.* goods, hotels), where x_i stands for the i^{th} review, the goal of the opinion summarization task is to generate a summary $y = \{w_1, w_2, \ldots, w_m\}$ of length m, which can convey the major aspect, sentiment polarity, and semantic of x_{raw}.

In this section, we introduce the implementation details of ConsistSum. The overall architecture is shown in Figure 2. In the first stage, ConsistSum extracts the aspect and sentiment distribution of each review x_i using the off-the-shelf ABAE [17] and TextCNN [20] model. In the second stage, we create the synthetic "review-summary" paired datasets. A sophisticated consistency metric function is devised to extract the preliminary datasets, and the consistency can be further improved by refining the raw pseudo-summary with the help of constrained sentence generation model CGMH [30]. In the last stage, we apply the teacher-forcing mechanism with additional loss of aspect and sentiment distribution to fine-tune the pre-trained summary model T5 [35], and apply beam search strategy to generate robust summaries.

3.1 Aspect and Sentiment Mining

The first step of our method is to conduct aspect and sentiment mining from the raw reviews. He *et al.* [17] have proposed the ABAE model, which can extract the aspect distribution with neural networks, achieving superior effectiveness and efficiency performance compared to traditional topic models. Following the ABAE model, we formalize the aspect mining as an unsupervised topic modeling process.

For each token w_i of the review $s = \{w_1, w_2, \cdots, w_k\}$, the word embedding $e_i \in \mathbb{R}^{d_e}$ can be obtained from embedding lookup matrix $M_e \in \mathbb{R}^{|V| \times d_e}$, where d_e is the embedding dimension and $|V|$ denotes the vocabulary size. Attention mechanism is used to generate the representation $z_s \in \mathbb{R}^{d_e}$ of the review s as follows:

$$\begin{aligned} z_s &= \sum_{i=1}^{k} a_i e_i \qquad a_i = softmax\left(d_i\right) \\ d_i &= e_i^T \cdot M_z \cdot y_s \qquad y_s = \frac{1}{k}\sum_{i=1}^{k} e_i \end{aligned} \tag{1}$$

where $y_s \in \mathbb{R}^{d_e}$ stands for the average of all word embeddings of review s. Besides, $M_z \in \mathbb{R}^{d_e \times d_e}$ is a parameter to be learned, which is used to weight the importance d_i of each word w_i and produce the review representation z_s. Moreover, from another perspective, we can obtain another review representation $r_s \in \mathbb{R}^{d_e}$ by aggregating aspects as follows:

$$\begin{aligned} r_s &= M_A^T \cdot p_a \\ p_a &= softmax\left(w_r \cdot z_s + b_r\right) \end{aligned} \tag{2}$$

where $M_A \in \mathbb{R}^{d_a \times d_e}$ is the aspect embedding matrix randomly initialized while d_a denotes the number of aspects. $w_r \in \mathbb{R}^{d_a \times d_e}$ and $b_r \in \mathbb{R}^{d_a}$ are linear parameters to transform z_s into the aspect distribution $p_a \in \mathbb{R}^{d_a}$ with softmax layer. Besides, the triple loss $Loss_{TRI}$ is incorporated to train ABAE, which expects to pull closer between the representation of z_s and r_s belonging to the same sample while alienating the representation n_i from different ones.

$$Loss_{TRI} = \sum_{s \in D} \sum_{i=1}^{\check{N}} \max\left(0, 1 - r_s z_s + r_s n_i\right) \tag{3}$$

where $\check{N}$ denotes the negative sample numbers of triple loss. Besides, the orthogonal regularization loss $Loss_{ORT}$ can help improve the uniqueness of the embedding of each aspect. Therefore, the final loss $Loss_{ABAE}$ consists of $Loss_{TRI}$ and $Loss_{ORT}$.

$$\begin{aligned} Loss_{ABAE} &= Loss_{TRI} + \lambda_{ORT} Loss_{ORT} \\ Loss_{ORT} &= \left\| M_A^T M_A - E \right\| \end{aligned} \tag{4}$$

where $E \in \mathbb{R}^{d_s}$ denotes the unit matrix and λ_{ORT} is the manual hyper parameter. At present, we are capable to obtain the aspect distribution $p_a \in \mathbb{R}^{d_a}$ via equation (2). Moreover, the sentiment distribution $p_s \in \mathbb{R}^{d_s}$ can be learned by TextCNN model [20] with the star ratings of reviews as supervision labels, where d_s denotes the sentiment dimension.

3.2 Consistency Training Dataset Creation

In this section, we conduct two steps to keep the consistency between review and summary in synthetic datasets.

3.2.1 Preliminary Dataset Extraction via Consistency Distance Metrics. In this section, we first define a novel consistency distance function to metric the correlation between reviews and summary, and then extract the closest raw review as pseudo-summary. The hit pseudo-summary should convey the major information of reviews including aspect, sentiment and semantic. Since the aspect and sentiment distribution have been introduced in Section 3.1, we just describe the method capturing semantic features here. Specifically, unsupervised sentence-level semantic representation has been extensively studied, including topic model [37], Sentence2Vector [32], pre-trained language model [40], and contrast learning [15]. In this paper, we leverage a simple yet effective approach by pooling the head and tail layer features of BERT [23]. The semantic representation Sem_{x_i} with respect to review x_i is as follows:

$$\begin{aligned} Sem_{x_i} &= POOL\left(\frac{Feat_{head} \oplus Feat_{tail}}{2}\right) \\ Feat_{head}, Feat_{tail} &= BERT\left(x_i\right) \end{aligned} \tag{5}$$

where $\oplus$ denotes the point-wise addition operation while $POOL(\cdot)$ averages the vector of words. Moreover, the consistency distance function $f_{dist}(x_i, x_j)$ *w.r.t.* review x_i and x_j includes three subparts: the semantic distance $f_{sem}(x_i, x_j)$ using cosine similarity, the aspect distance $f_{asp}(x_i, x_j)$ and the sentiment distance $f_{senti}(x_i, x_j)$ measured by Jensen-Shannon (JS) divergence.

$$\begin{aligned} f_{dist}\left(x_i, x_j\right) =& \lambda_{sem} f_{sem}\left(x_i, x_j\right) + \lambda_{asp} f_{asp}\left(x_i, x_j\right) + \lambda_{senti} f_{senti}\left(x_i, x_j\right) \\ f_{sem}\left(x_i, x_j\right) =& 1 - \frac{1 - cos\left(x_i, x_j\right)}{2} \qquad cos\left(x_i, x_j\right) = \frac{Sem_{x_i} \cdot Sem_{x_j}}{\left|Sem_{x_i}\right| \times \left|Sem_{x_j}\right|} \\ f_{asp}\left(x_i, x_j\right) =& 1 - \left(\frac{1}{2} KL\left(p_{a_i} \| \frac{p_{a_i} + p_{a_j}}{2}\right) + \frac{1}{2} KL\left(p_{a_j} \| \frac{p_{a_i} + p_{a_j}}{2}\right)\right) \\ f_{senti}\left(x_i, x_j\right) =& 1 - \left(\frac{1}{2} KL\left(p_{s_i} \| \frac{p_{s_i} + p_{s_j}}{2}\right) + \frac{1}{2} KL\left(p_{s_j} \| \frac{p_{s_i} + p_{s_j}}{2}\right)\right) \end{aligned} \tag{6}$$

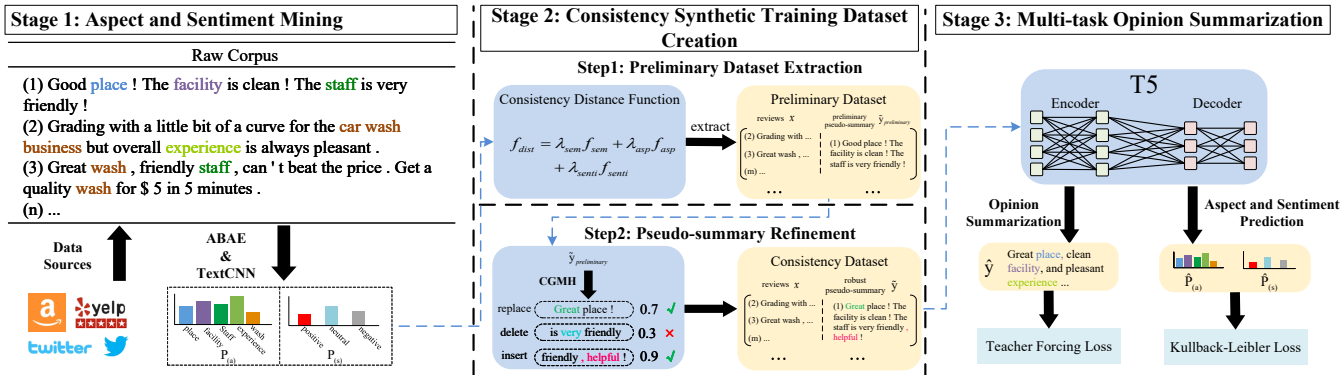


Figure 2: Overview of ConsistSum workflow. The dotted arrow indicates the data flow from previous stage to next one.

where $\lambda_{sem}, \lambda_{asp}, \lambda_{senti}$ are manually set coefficients and subject to the condition $\lambda_{sem} + \lambda_{asp} + \lambda_{senti} \equiv 1$. $p_{a_i} \in \mathbb{R}^{d_a}$ and $p_{a_j} \in \mathbb{R}^{d_a}$ denote the aspect distributions corresponding to the reviews x_i and x_j, respectively. Similarly, $p_{s_i} \in \mathbb{R}^{d_s}$ and $p_{s_j} \in \mathbb{R}^{d_s}$ represent the sentiment distribution. $KL(\cdot)$ denotes Kullback-Leibler (KL) divergence and we can calculate JS divergence via KL divergence to maintain the symmetry of inputs. More, we conduct some transformation on cosine similarity and JS divergence, *e.g.*, 1 minus JS divergence, because the consistency distance should be normalized to the positive correlation interval of [0,1], *i.e.*, the larger $f_{dist}(x_i, x_j)$ indicates the closer distance of (x_i, x_j).

Under the direction of consistency distance function, we can extract the preliminary "review-summary" dataset. Specifically, given the raw corpus $x_{raw} = \{x_1, x_2, \ldots, x_n\}$, previous studies mainly first choose a review as a preliminary pseudo-summary $\tilde{y}_{preliminary}$, and then extract corresponding reviews via similarity calculation. Such greedy process would select synthetic reviews close to the pseudo summary, resulting in insufficient diversity. However, reviews are often diverse in reality. For example, for a restaurant, some reviewers describe the environment, while others might talk about the service. To ensure the diversity the synthetic of datasets, we employ the opposite strategy. Firstly, A review set x containing N reviews is randomly selected, where N is a hyper-parameter identifying the number of reviews per sample. Secondly, we calculate the center x_{center} of selected reviews, which includes the feature of aspect, sentiment and semantic. Thirdly, the closest review x_{close} is hit within all the consistency distances between unselected reviews and x_{center}. Eventually, if the distance $f_{dist}(x_{close}, x_{center})$ is greater than the manually set threshold δ_{dist}, x_{close} will be regarded as the preliminary pseudo-summary $\tilde{y}_{preliminary}$. Otherwise, the process will jump to the first step until the satisfying sample $< x, \tilde{y}_{preliminary} >$ constructed.

3.2.2 Pseudo-Summary Refinement via Constrained Sentence Generation. When a person writes a review, the initial intention is to express his/her opinion rather than to cover opinions of other people. Therefore, there is still a certain gap between the preliminary reviews x and pseudo-summary $\tilde{y}_{preliminary}$ since both of them come from the raw reviews. However, current methods ignore this reality when constructing synthetic datasets [1, 2, 43]. To tackle this issue, we set the closeness of consistency distance with reviews x as constrains, and apply CGMH [30, 38], a representative constrained sentence generation model, to fine-tune the pseudo-summary with specific edit operations, *i.e.*, insertion, deletion, replacement. Such edit operations are conducted by sampling from pre-trained language model (LM) BERT iteratively. Formally, we first define the joint probability distribution $\pi(y, c)$ of LM and constrains.

$$\pi(y, c) = p(y) \cdot \varphi(y, c) \tag{7}$$

where $p(y)$ denotes the probability of BERT when generating y, and c stands for the sampling constraints. $\varphi(y, c)$ is an indicative function with hard constraints, *i.e.*, when c meets the constraint, $\varphi(y, c) = 1$, otherwise $\varphi(y, c) = 0$. Based on this, we can have the conditional probability distribution $p(y|c)$ as follows:

$$p(y \mid c) = \frac{\pi(y, c)}{\sum_y \pi(y, c)} \tag{8}$$

We expect that sampling from $p(y|c)$ can boost the preliminary pseudo-summary to be closer to the review set x. However, sampling from $p(y|c)$ directly is challenging since we can not calculate $\sum_y \pi(y, c)$ intuitively. Fortunately, Metropolis-Hasting (M-H) method, a variant of Markov Chain Monte Carlo (MCMC), can sample step-by-step instead of directly sampling from complicated distribution. It conforms to our idea of "fine-tuning" rather than "regenerating" towards the pseudo-summary $\tilde{y}$. In the concrete implementation, we first define the acceptance probability $ACC\left(\tilde{y}_i \rightarrow \tilde{y}_{i+1}\right)$ for M-H sampling process.

$$ACC\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right) = \min\left(1, \frac{q\left(\tilde{y}_i \leftarrow \tilde{y}_{i+1}\right) p\left(\tilde{y}_{i+1}\right)}{q\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right) p\left(\tilde{y}_i\right)}\right) \tag{9}$$

where $q\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right)$ denotes the probability that i^{th} step pseudo-summary $\tilde{y}_i$ transfers into $(i+1)^{th}$ step pseudo-summary $\tilde{y}_{i+1}$ using edit operations, and $q\left(\tilde{y}_i \leftarrow \tilde{y}_{i+1}\right)$ vice versa. Specifically, the replacing transition probability $q_{replace}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right)$ can be considered as BERT's generation probability of $\tilde{y}_{i+1}$ when masking the word $\tilde{y}_i^z$ in $\tilde{y}_i$.

$$\begin{aligned} q_{replace}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right) = p_z \cdot BERT\Big(&\left\{\tilde{y}_{i+1}^1, \cdots, \tilde{y}_{i+1}^z, \cdots\right\} \\ &\leftarrow \left\{\tilde{y}_i^1, \cdots, \tilde{y}_i^{z-1}, [MASK], \tilde{y}_i^{z+1}, \cdots\right\}\Big) \end{aligned} \tag{10}$$

where p_z is the random probability of $\tilde{y}_i^z$ being selected. Moreover, the insertion operation can be understood as randomly selecting a position of $\tilde{y}_i$ and filling with $[MASK]$ firstly, then replacing $[MASK]$ with the corresponding token of $\tilde{y}_{i+1}$. Therefore, the insertion transition probability $q_{insert}\left(\tilde{y}_i \leftarrow \tilde{y}_{i+1}\right)$ can be obtained via $q_{replace}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right)$. Analogously, the deletion operation can be processed as directly deleting the token $\tilde{y}_i^z$ of the random selected position. Thus, we can define $q_{delete}\left(\tilde{y}_i \leftarrow \tilde{y}_{i+1}\right) = p_z$.

Besides, M-H judges whether to accept each sampling process through the acceptance rate. Here, replacement itself and insertion-deletion are inverse operations, and the acceptance rate ACC is defined as follows:

$$\begin{aligned}
ACC_{replace}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right) &= \frac{p_{replace} \cdot p\left(\tilde{y}_{i+1} \mid c\right) \cdot q_{replace}\left(\tilde{y}_i \leftarrow \tilde{y}_{i+1}\right)}{p_{replace} \cdot p\left(\tilde{y}_i \mid c\right) \cdot q_{\text{replace}}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right)} \\
&= \frac{p_{replace} \cdot \pi\left(\tilde{y}_{i+1}, c\right) \cdot q_{\text{replace}}\left(\tilde{y}_i \leftarrow \tilde{y}_{i+1}\right)}{p_{replace} \cdot \pi\left(\tilde{y}_i, c\right) \cdot q_{replace}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right)} \\
ACC_{insert}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right) &= \frac{p_{delete} \cdot p\left(\tilde{y}_{i+1} \mid c\right) \cdot q_{delete}\left(\tilde{y}_i \leftarrow \tilde{y}_{i+1}\right)}{p_{insert} \cdot p\left(\tilde{y}_i \mid c\right) \cdot q_{insert}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right)} \\
&= \frac{p_{delete} \cdot \pi\left(\tilde{y}_{i+1}, c\right) \cdot q_{delete}\left(\tilde{y}_i \leftarrow \tilde{y}_{i+1}\right)}{p_{insert} \cdot \pi\left(\tilde{y}_i, c\right) \cdot q_{insert}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right)} \\
ACC_{delete}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right) &= \frac{p_{insert} \cdot p\left(\tilde{y}_{i+1} \mid c\right) \cdot q_{insert}\left(\tilde{y}_i \leftarrow \tilde{y}_{i+1}\right)}{p_{delete} \cdot p\left(\tilde{y}_i \mid c\right) \cdot q_{delete}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right)} \\
&= \frac{p_{insert} \cdot \pi\left(\tilde{y}_{i+1}, c\right) \cdot q_{insert}\left(x \leftarrow \tilde{y}_{i+1}\right)}{p_{delete} \cdot \pi\left(\tilde{y}_i, c\right) \cdot q_{delete}\left(\tilde{y}_{i+1} \leftarrow \tilde{y}_i\right)}
\end{aligned} \tag{11}$$

where $p_{replace} \in \mathbb{R}^3$, $p_{insert} \in \mathbb{R}^3$, $p_{delete} \in \mathbb{R}^3$ indicate the edit operation's probability in procedure, which are set manually. The indicative function $\varphi(y, c)$ can control the consistency improvement in process, which is defined as follows:

$$\varphi(y, c) = \begin{cases} 0, & f_{dist}\left(\tilde{y}_{i+1}, x_{center}\right) \leq f_{dist}\left(\tilde{y}_i, x_{center}\right) \\ 1, & f_{dist}\left(\tilde{y}_{i+1}, x_{center}\right) > f_{dist}\left(\tilde{y}_i, x_{center}\right) \end{cases} \tag{12}$$

where x_{center} represents the center of reviews x. To sum up, we can conduct CGMH with $step_{num}$ times to produce more robust pseudo-summary $\tilde{y}$, and pair $< x, \tilde{y} >$ as training datasets.

3.3 Multi-task Opinion Summarization Model Training and Inference

We first utilize NLTK [5] to filter stop words and stitch the review segments as inputs. In the training stage, we fine-tune the pre-trained generative model T5 [35] with teacher-forcing strategy. In the inference stage, we employ beam search to produce more robust summaries. Due to the large-scale vocabulary of T5, we apply the sparse softmax function [29, 39] to allow the generated token distribution more concentrated.

$$Sparse\_Softmax(p) = \begin{cases} \frac{e^{p_i}}{\sum_{j \in \Phi_k} e^{p_j}}, & i \in \Phi_k \\ 0, & i \notin \Phi_k \end{cases} \tag{13}$$

where Φ_k denotes the subscripts of top k tokens with highest probability, and k is a hyper parameter.

Moreover, in order to better adapt T5 to the opinion summarization task, we introduce a bypass branch when decoding, which aim to infer the aspect and sentiment distribution of the generated summary. Under this design, we incorporate a novel consistency loss $Loss_{Consisit}$ as KL divergence of the reviews and summary distributions. In a nutshell, T5 would be fine-tuned with the multi-task schema, and the training loss $Loss_{Summary}$ is as follows.

$$\begin{aligned}
Loss_{Summary} &= Loss_{Seq2Seq} + \lambda_{consist} Loss_{Consisit} \\
Loss_{Consisit} &= \lambda_a KL\left(p_a \| \widehat{p_a}\right) + \left(1 - \lambda_a\right) KL\left(p_s \| \widehat{p_s}\right)
\end{aligned} \tag{14}$$

where $Loss_{Seq2Seq}$ denotes the sequence-to-sequence loss of T5. $\lambda_{consist}$ and λ_a are hyper parameters to be set manually. $p_a \in \mathbb{R}^{d_a}$ and $p_s \in \mathbb{R}^{d_s}$ denote the aspect and sentiment distribution of reviews respectively, which can be mined in Section 3.1. Besides, $\widehat{P_a} \in \mathbb{R}^{d_a}$ and $\widehat{P_s} \in \mathbb{R}^{d_s}$ denote the aspect and sentiment distribution of generated summary. We can obtain them using the T5 hidden state with a fully connected and softmax normalization layer.

4 EXPERIMENTS

In this section, we first introduce the benchmark datasets and experimental settings. Then, we briefly describe two type representative baselines, *i.e.*, extractive and abstractive methods. Moreover, we report and analyze the results of the main experiment and ablation experiments. Finally, further case studies are provided to demonstrate the effectiveness of our proposed method.

4.1 Dataset and Experimental Settings

We conducted experiments on the Yelp dataset [9] and Amazon dataset [7]. The Yelp dataset is a large-scale dataset lacking gold-standard summaries. Previous studies have created the small-scale validation and test datasets by the amazon crowdsourcing platform (AMT), where each summary corresponds to 8 reviews. For the Amazon dataset, its topics include electronics, clothing, health advice, *etc.* Similarly, this dataset does not deliver reference summaries and the validation and test samples are also provided through the AMT platform. Different from the Yelp dataset, the AMT platform reports three reference summaries for each sample for the Amazon dataset. Besides, each review of both datasets contains a specific discrete rating label from 1 to 5, which conveys the reviewer's sentiment polarity. Finally, we create the synthetic dataset for training, where each summary also corresponds to 8 reviews. The detail information of datasets is shown in TABLE 1.

Table 1: The statistic of the two datasets

Dataset	Raw Corpus Reviews	Synthetic Dataset	Dev	Test
Yelp	2320800	95000	100	100
Amazon	1175191	85000	28	32

In this paper, we implement ConsistSum by PyTorch 1.8.1 with four NVIDIA TESLA P100 platforms, and each platform is equipped four 16G GPUs. Besides, several open source models are also involved: GloVe 42B embedding [33] for aspect and sentiment distribution extraction, BERT base version for CGMH sampling, T5 base version for training the summarization model, and NLTK toolkit for filtering stop words.

As introduced in Section 3 , some hyper parameters would influence the performance. In the experiment, we set the adjustment parameters λ_{ORT}, λ_{sem}, λ_{asp}, λ_{senti}, $\lambda_{consist}$, λ_a into 0.1, 0.15, 0.5, 0.35, 0.2, 0.8, respectively. In the first stage, we set the dimension

$d_a = 30$ and $d_s = 5$ w.r.t. aspect and sentiment distribution respectively. The negative sample numbers of triple loss $\check{N}$ is set to 10. In the synthetic dataset creation stage, the consistency distance threshold δ_{dist} is set to 0.75 and the CGMH sampling times $step_{num}$ is set to 30. In addition, we set the edit operation probability as: $p_{replace} = 0.6$, $p_{insert} = p_{delete} = 0.2$, where the relatively higher replacement probability encourages to keep the length of original pseudo-summary. In the summary generation stage, we set the valid sparse softmax number k=300, the max sentence length $max_{len} = 150$ and the beam search size into 5. Finally, we use the Adam [21] optimizer to learn the model parameters and incorporate the L2-regularization with $\lambda = 0.01$ to avoid over-fitting. The learning rate lr is set to $1e-3$ for ABAE and TextCNN, and $2e-4$ for T5. The batch size is 32, 32 and 8, and the epoch number is 10, 10 and 5 for ABAE, TextCNN and T5, respectively.

4.2 Baseline Methods

To evaluate the performance of ConsistSum, representative unsupervised extractive and abstractive opinion summarization models are chosen as our baselines.

4.2.1 Extractive Models. **LexRank** [11] is a graph algorithm using TF-IDF to calculate weights of sentence segments and select segments near center as the summary. **W2vCent** [36], **SnCent** [2], and **BertCent** [1] first calculate the sentence-level semantics centroids with Word2Vec, LSTM language model and BERT, respectively. Then, they greedily extract reviews that are close to these centroids.

4.2.2 Abstractive Models. **Opinosis** [13] is the early abstractive model which applies the structure of graph to eliminate redundancy and generate the summary. **MeanSum** [9] applies the auto-encoder model with self-reconstruction loss to learn the feature of reviews, and aggregates review features to produce the corresponding summary. Taking MeanSum [9] a step further, **CopyCat** [7] applies variational auto-encoder (VAE) to capture richer hidden state of reviews and summary. **DenoiseSum** [2] first creates the synthetic dataset by injecting specific noise into the pseudo-summary, and then trains the opinion summarization model with an exquisite denoising mechanism. **OpinionDigest** [41] expects the opinion span can be capable to reconstruct the original review, and establishes the sequence-to-sequence samples based on this idea. **PlanSum** [1] incorporates the content planning induction to extract the synthetic datasets from raw corpus, and trains the opinion summarization model under supervision.

Finally, we leverage the ROUGE score [26] to evaluate the quality summary. In this paper, ROUGE-1 (R1), ROUGE-2 (R2), ROUGE-L (RL) are utilized to evaluate the matching patterns of 1-gram, 2-gram, and longest common sub-sequence, respectively.

4.3 Main Results

The experiment results are presented in Table 2. It shows that the BertCent model performs best in the extractive method group. Regarding R1, R2, and RL, it achieves 26.67%, 3.19% and 14.67% on Yelp dataset and 30.67%, 5.21%, and 17.76% on Amazon dataset, respectively. Meanwhile, except for our model ConsistSum, the PlanSum model works best in the abstractive method group. Regarding R1, R2, and RL, it achieves 34.79%, 7.01% and 19.74% on Yelp dataset and 32.87%, 6.12%, and 19.05% on Amazon dataset, respectively. ConsistSum outperforms all state-of-the-art models under the measurement of R2 and RL on Yelp dataset. On Amazon dataset, ConsistSum outperforms all state-of-the-art models regarding the R1 and RL. For the R1 on Yelp dataset and R2 on Amazon dataset, the PlanSum model achieves the highest score; however, ConsistSum obtains acceptable results, which are only lower than PlanSum around 2% on Yelp and 0.2% on Amazon. The result indicates ConsistSum can maintain a good performance across different datasets under various measures.

Table 2: Performance of different methods on the two datasets. The best results are in bold while the second best ones are underlined. The results with †, ‡ and ♮ are from the experiments of Arthur *et al.* [7], Reinald *et al.* [1] and Yoshihiko *et al.* [41], respectively.

Model	Yelp			Amazon		
	R1 (%)	R2 (%)	RL (%)	R1 (%)	R2 (%)	RL (%)
LexRank	25.50‡	2.64‡	13.37‡	28.74‡	5.47‡	16.75‡
W2vCent	24.61‡	2.85‡	13.81‡	28.73‡	4.97‡	17.45‡
SnCent	25.05‡	3.09‡	14.56‡	30.45‡	5.40‡	17.73‡
BertCent	26.67‡	3.19‡	14.67‡	30.67‡	5.21‡	17.76‡
Opinosis	25.15‡	2.61‡	13.54‡	28.42‡	4.57‡	15.50‡
MeanSum	28.86‡	3.66‡	15.91‡	29.20‡	4.70‡	18.15‡
CopyCat	29.47†	5.26†	18.09†	31.97†	5.81†	<u>20.16</u>†
DenoiseSum	30.14‡	4.99‡	17.65‡	—	—	—
OpinionDigest	29.30♮	5.77♮	18.56♮	—	—	—
PlanSum	**34.79**‡	<u>7.01</u>‡	<u>19.74</u>‡	<u>32.87</u>‡	**6.12**‡	19.05‡
ConsistSum	<u>32.65</u>	**7.49**	**20.87**	**33.32**	<u>5.94</u>	**21.41**

It is obvious that the performance of abstractive methods is generally better than extractive ones. This might profit from that the abstractive model can absorb the whole compressed review features to generate the summary, which can alleviate the information loss of extractive models. Meanwhile, abstractive models can obtain a more fluent abstract with the help of pre-trained language models.

Another observation is that the models utilized the sequence-to-sequence schema on the synthetic dataset, such as PlanSum, OpinionDigest, and DenoiseSum, exhibit superior performance than the auto-encoder models, *i.e.*, MeanSum and CopyCat. Such result demonstrates that the sequence-to-sequence strategy is more suitable for generating the robust opinion summary. Even so, ConsistSum still outperforms existing sequence-to-sequence based models to a certain extent in most cases. The major reason is that we devise the complete mechanism to improve the consistency in each process of ConsistSum. Moreover, compared to other abstractive methods, ConsistSum emphasizes more information on the aspect and sentiment besides semantic.

4.4 Ablation Experiments

We design the three components of ConsistSum to ensure the consistency between reviews and corresponding summary, including consistency distance function, edit operations with CGMH, and multi-task (MT) learning schema. To evaluate the effectiveness

Table 3: Results of the main components ablation experiment. MT denotes the multi-task learning schema of T5.

Model	Yelp		Amazon	
	RL (%)	Decline ↓	RL (%)	Decline ↓
ConsistSum	20.87	-	21.41	-
w/o aspect	19.84	1.03	19.82	1.59
w/o sentiment	20.13	0.74	20.21	1.2
w/o semantic	20.3	0.57	20.73	0.68
w/o CGMH	19.75	1.12	19.26	2.15
w/o MT	20.69	0.18	21.08	0.33

of each component, we conduct the ablation experiment and the results are shown in Table 3.

4.4.1 About Consistency Distance Function. The consistency distance function includes features of aspect, sentiment and semantic. The performance decline is: ConsistSum w/o aspect > ConsistSum w/o sentiment > ConsistSum w/o semantic. The performance ranking reveals that rather than semantic, aspect and sentiment should be taken priority of.

4.4.2 About CGMH. Compared to other variants, ConsistSum w/o CGMH achieves the worst performance. This observation reflects the defect of preliminary datasets directly extracted from raw corpus without modification. It also demonstrates that CGMH can substantially improve the consistency by processing edit operations progressively.

4.4.3 About Multi-task Learning Schema. The performance of ConsistSum w/o MT drops slightly in Table 3. On the one hand, it indicates that incorporating the aspect and sentiment prediction task can boost the performance. On the other hand, the little decline might result from the rich semantic features of T5, which could make up the aspect and sentiment information to some extent.

4.5 Case Study

In this section, we conduct two visual case studies, *i.e.*, summarization results visualization and CGMH sampling process.

4.5.1 Summarization Results Visualization. We choose PlanSum and CopyCat, two representative state-of-the-art abstractive methods, as baselines to visualize and compare the summaries. Besides, ConsistSum w/o MT is also included in order to analyze the effect of multi-task learning schema. Figure 3 shows the generated summaries, where different colors stand for different aspects. Besides, the sign "~" denotes the prediction result with wrong sentiment polarity. Besides, words underlined by the sign "_" represent the missing aspects. Due to space constraints, parts of the reviews are omitted. In short, the original reviews includes four aspects of a car wash shop: (1) "service" with positive polarity, (2) "price" with slight negative polarity, (3) "efficiency" with positive polarity, and (4) "quality" with positive polarity.

In brief, baseline models have following flaws: (1) PlanSum inexplicably mentions the "food" aspect that does not exist in reviews, and the sentiment of price is incorrectly predicted to positive. (2) The summary generated by CopyCat is short, which only covers the "service" and "efficient" aspects, resulting in the omission of significant information. (3) ConsistSum w/o MT incorrectly predicts the sentiment polarity of the "price" and "quality" of car wash (*e.g.*, reasonably price, unclean washing). Comparatively, the summary produced by ConsistSum can completely cover all aspects with correct sentiment polarities. The conclusion is that ConsistSum can still protect the consistency of summary and reviews when dealing with complicated reviews with multiple aspects and different sentiment polarities on each aspect.

Model	Summary
PlanSum	This is a great place to go for a quick bite to eat. The staff is very friendly and helpful. They have a lot of options to choose from and *the prices are very reasonable*. I would recommend this place if you're in the area and want a good car wash. **clean**
CopyCat	If you're looking for a quick car wash, this is the place to go! The staff is always friendly and helpful. I've been going here for years and will continue to go back! **Price, clean ·····**
ConsistSum w/o MT	This is a great place to get a wash and vacuum. The staff is friendly and the wash is quick and easy. *The prices are reasonable* and the wash is quick. The only thing I would say is that *the wash is not as clean as the other places*.
ConsistSum	This is a great place to get a wash and vacuum. The staff is always friendly and helpful. They have a lot of opinions to choose and the wash is quick and clean. The only thing I would say is that the prices are a little high than other places.

Figure 3: Four generated summaries of PlanSum, CopyCat, ConsistSum w/o MT and ConsistSum, where different colors stand for different aspects. The sign "~" denotes the predictive segments with wrong sentiment polarity and the missing aspects are underlined by "_".

4.5.2 CGMH Sampling Procedure. Figure 4 illustrates the sampling procedure of CGMH with constraints. Each row represents an edit operation on the pseudo-summary.

Pseudo-summary:
They are amazing here! They could not have done a more perfect job on my eyebrows! They have this warm inviting feeling when you walk into their business. They make you feel as if you've been coming there for years and you're apart of the family. We also got eyelashes extentions there and the girl who did my eyelashes can work her magic! You cannot even tell I have extentions because she did such a good job! Highly recommend!

Step	CGMH Sampling State	Edit Operation	Consistency Distance	Accept /Reject
0	*Initial state*	-	0.8845	-
1	They are truly amazing here! …	Insertion	0.9079	Accept
2	… I also got eyelashes extentions there and the girl who did my eyelashes can work her magic! …	Replacement (We → I)	0.9148	Accept
3	… I also got eyelashes extentions there and the girl who did my eyelashes can show her magic! …	Replacement (work → show)	0.9176	Accept
4	… I also got eyelashes extentions there and the little girl who did my eyelashes can work magic! …	Insertion	0.8711	Reject
5	…highly recommend! …	Deletion	0.8637	Reject
6	… They have this warm inviting service when you walk into their business. …	Replacement (feeling → service)	0.9305	Accept

Figure 4: Six representative sampling steps of CGMH, which conducts the edit operation in each step.

Only the step that shortens the distance between pseudo-summary and reviews can be accepted. In the first step, the word "truly" enhances the sentiment polarity. In the second step, the replacement of "We" with "I" enables the consistency with the subsequent word "my". The third step can lead to more fluent of syntax, because we can get a more common expression "show her magic" by replacing "work" with "show". In the last step, replacing "feeling" with

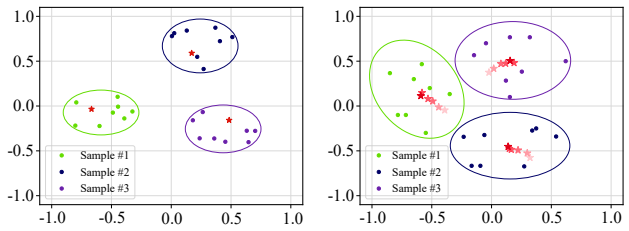
"service" makes the aspect feature more concentrated and improves the aspect consistency towards reviews. Moreover, we analyze the reason of rejected edit operations. The insertion of "little" in the 4^{th} step, turning "the girl" into "the little girl", may allow the grammar more coherent but make the pseudo-summary deviating from the correct meaning. In the 5^{th} step, "highly" is deleted, which weakens the sentiment polarity of the reviews. To sum up, by constantly accepting or rejecting samples, CGMH can continuously polish the pseudo-summary to improve its consistency with reviews.

5 DISCUSSION

In this section, we further discuss the effect of synthetic dataset diversity and CGMH sampling.

5.1 Effect of Synthetic Dataset Diversity

The diversity of synthetic datasets created by ConsistSum and PlanSum are compared in this section. For both abstractive methods, we randomly choose three samples and map the aspect distributions into two-dimensional space via principal component analysis (PCA), because the aspect is the most significant feature through Section 4.4. Figure 5(a) plots the PlanSum's result of the three samples. We can observe that the aspects of reviews are too concentrated, and corresponding pseudo-summary would deviate from the center of reviews. In contrast, Figure 5(b) shows that ConsistSum can create the synthetic datasets with high diversity of reviews. Moreover, the procedure of the pentagram from the virtue to the real in Figure 5(b) represents the evolution of the pseudo-summary using CGMH sampling. As CGMH sampling continues, the pseudo-summary moves closer to the center of reviews. In conclusion, the above analysis demonstrates that ConsistSum shows superiority in generating synthetic datasets with higher diversity and consistency.

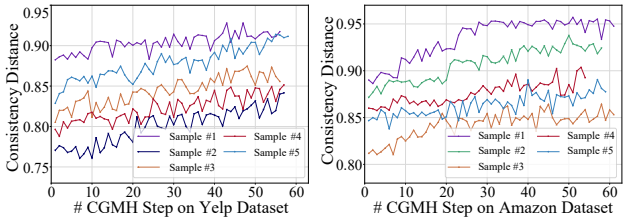


(a) PlanSum (b) ConsistSum

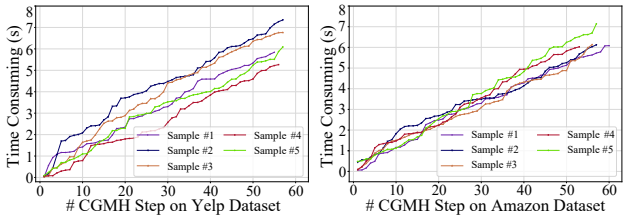
Figure 5: The two-dimensional aspect distribution regarding three samples from synthetic datasets. The dot sign indicates review distributions, and the pentagram sign indicates the distribution of pseudo-summaries.

5.2 Effect of CGMH Sampling

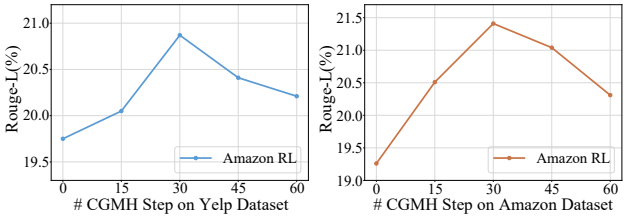
To verify the effectiveness of CGMH, we adjust its sampling times $step_{size}$ from 1 to 60 and analyze the evolution of performance. Firstly, Figure 6(a) shows that with the increase of sampling steps, the consistency distance increases continuously because we only accept the steps closing the consistency distance. Meanwhile, Figure 6(b) shows that the time consumed increases linearly with the iteration of sampling. Therefore, multiple GPUs with parallel sampling are required to conduct CGMH. Moreover, we can observe in Figure 6(c) that for Yelp and Amazon Datasets, CGMH's sampling step size shows a concave shape with ConsistSum's performance rather than an absolute positive correlation, and ConsistSum achieves the best performance when $step_{size} = 30$. The possible reason is that the CGMH constraints employ the consistency distance function, which incorporates three hyper parameters to compute the weights of aspect, sentiment, and semantic, respectively. Such hyper parameters might introduce deviations from the genuine consistency.



(a) The consistency distance and acceptance step.



(b) The time Consuming.



(c) The ConsistSum performance.

Figure 6: Effect illustration of CGMH sampling steps. (a) shows the correlation between consistency distance and CGMH sample step. (b) shows the time consuming of CGMH. (c) shows the ConsistSum performance under the settings of different CGMH sampling steps.

6 CONCLUSION

In this paper, we analyze the main challenges of the unsupervised opinion summarization task and propose a novel consistency opinion summarization method (ConsistSum). Our work devotes to control the consistency of aspect, sentiment and semantic between reviews and summary. Experimental results show ConsistSum can create more robust synthetic datasets and generate more comprehensive summaries, thus outperforming state-of-the-art baselines.

7 ACKNOWLEDGMENTS

This paper is funded by the National Natural Science Foundation of China under Grant Nos. 62002347, U21B2046 and 91746301. Huawei Shen is also supported by Beijing Academy of Artificial Intelligence (BAAI) under the grant number BAAI2019QN0304 and K.C. Wong Education Foundation.

REFERENCES

[1] Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Unsupervised Opinion Summarization with Content Planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 12489–12497.

[2] Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised Opinion Summarization with Noising and Denoising. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), 1934–1945.

[3] Reinald Kim Amplayo and Mirella Lapata. 2021. Informative and Controllable Opinion Summarization. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (2021), 2662–2672.

[4] Stefanos Angelidis and Mirella Lapata. 2018. Summarizing Opinions: Aspect Extraction Meets Sentiment Prediction and They Are Both Weakly Supervised. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (2018), 3675–3686.

[5] Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. 69–72.

[6] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Few-Shot Learning for Opinion Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 4119–4135.

[7] Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Unsupervised Opinion Summarization as Copycat-Review Generation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), 5151–5169.

[8] Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2020. CoCon: A Self-Supervised Approach for Controlled Text Generation. In *International Conference on Learning Representations*.

[9] Eric Chu and Peter Liu. 2019. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*. PMLR, 1223–1232.

[10] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. *International Conference on Learning Representations* (2019).

[11] Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22 (2004), 457–479.

[12] Xianghua Fu, Yanzhi Wei, Fan Xu, Ting Wang, Yu Lu, Jianqiang Li, and Joshua Zhexue Huang. 2019. Semi-supervised aspect-level sentiment classification model based on variational autoencoder. *Knowledge-Based Systems* 171 (2019), 81–92.

[13] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. 340–348.

[14] René Arnulfo García-Hernández, Romyna Montiel, Yulia Ledeneva, Eréndira Rendón, Alexander Gelbukh, and Rafael Cruz. 2008. Text summarization by sentence extraction using unsupervised learning. In *Mexican International Conference on Artificial Intelligence*. Springer, 133–143.

[15] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 879–895.

[16] Lu He, Tingjue Yin, Zhaoxian Hu, Yunan Chen, David A Hanauer, and Kai Zheng. 2021. Developing a standardized protocol for computational sentiment analysis research using health-related social media data. *Journal of the American Medical Informatics Association* 28, 6 (2021), 1125–1134.

[17] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 388–397.

[18] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 168–177.

[19] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT* (2019), 4171–4186.

[20] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing(EMNLP)*. 1746–1751.

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *Computer Science* (2014).

[22] Daniel Lee, Rakesh Verma, Avisha Das, and Arjun Mukherjee. 2020. Experiments in Extractive Summarization: Integer Linear Programming, Term/Sentence Scoring, and Title-driven Models. *arXiv e-prints* (2020), arXiv–2008.

[23] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[24] Qintong Li, Piji Li, Xinyi Li, Zhaochun Ren, Zhumin Chen, and Maarten de Rijke. 2021. Abstractive Opinion Tagging. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 337–345.

[25] Yingjie Li and Cornelia Caragea. 2019. Multi-task stance detection with sentiment and stance lexicons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 6299–6305.

[26] Chin-Yew Lin and Eduard Hovy. 2003. 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*. 150–157.

[27] Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, Tag, Realize: High-Precision Text Editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5054–5065.

[28] Yuning Mao, Wenchang Ma, Deren Lei, and Xiang Ren. 2021. Extract, Denoise, and Enforce: Evaluating and Predicting Lexical Constraints for Conditional Text Generation. *arXiv e-prints* (2021), arXiv–2104.

[29] André FT Martins and Ramón F Astudillo. 2016. From softmax to sparsemax: a sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*. 1614–1623.

[30] Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6834–6842.

[31] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[32] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 528–540.

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

[34] Jiangtao Qiu, Chuanhui Liu, Yinghong Li, and Zhangxi Lin. 2018. Leveraging sentiment analysis at the aspects level to predict ratings of reviews. *Information Sciences* 451 (2018), 295–309.

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21 (2020), 1–67.

[36] Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. 2017. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*. 12–21.

[37] Vivek Kumar Rangarajan Sridhar. 2015. Unsupervised topic modeling for short texts using distributed representations of words. In *Proceedings of the 1st workshop on vector space modeling for natural language processing*. 192–200.

[38] Jianlin Su. 2021. Make sentences by adding, deleting and replacing words. https://spaces.ac.cn/archives/8194

[39] Jianlin Su. 2021. SPACES: long document summarization with extraction and abstraction operations. https://spaces.ac.cn/archives/8046

[40] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening Sentence Representations for Better Semantics and Faster Retrieval. *arXiv e-prints* (2021), arXiv–2103.

[41] Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. 2020. OpinionDigest: A Simple Framework for Opinion Summarization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), 5789–5798.

[42] Wenyi Tay. 2019. Not All Reviews Are Equal: Towards Addressing Reviewer Biases for Opinion Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. 34–42.

[43] Ke Wang and Xiaojun Wan. 2021. TransSum: Translating Aspect and Sentiment Embeddings for Self-Supervised Opinion Summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 729–742.

[44] Yumo Xu and Mirella Lapata. 2020. Coarse-to-fine query focused multi-document summarization. In *Proceedings of the 2020 Conference on empirical methods in natural language processing (EMNLP)*. 3632–3645.

[45] Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and William B Dolan. 2020. Pointer: Constrained Text Generation via Insertion-based Generative Pre-training. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2020), 8649–8670.

[46] Chao Zhao and Snigdha Chaturvedi. 2020. Weakly-supervised opinion summarization by leveraging external information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9644–9651.