Opinion Summarization by Weak-Supervision from Mix-structured Data

Yizhu Liu[1] **Qi Jia**[2] **Kenny Q. Zhu**[3*]
Shanghai Jiao Tong University
Shanghai, China
{[1]liuyizhu, [2]Jia_qi}@sjtu.edu.cn, [3]kzhu@cs.sjtu.edu.cn

Abstract

Opinion summarization of multiple reviews suffers from the lack of reference summaries for training. Most previous approaches construct multiple reviews and their summary based on textual similarities between reviews, resulting in information mismatch between the review input and the summary. In this paper, we convert each review into a mix of structured and unstructured data, which we call opinion-aspect pairs (OAs) and implicit sentences (ISs). We propose a new method to synthesize training pairs of such mix-structured data as input and the textual summary as output, and design a summarization model with OA encoder and IS encoder. Experiments show that our approach outperforms previous methods on Yelp, Amazon and RottenTomatos datasets.

1 Introduction

Opinion summarization aims at producing a summary for a set of subjective user reviews about an entity (i.e., a product, a service or a movie), which can help users quickly understand the entity. Opinion summarization is a specialized multi-document summarization (Fabbri et al., 2019; Liu et al., 2018a; Ma, 2021). Unlike traditional multi-document summarization, opinion summarization focuses on the aspects and opinions of entities in the input documents. Deep learning techniques have made great successes in summarization (See et al., 2017; Liu et al., 2018b; Lewis et al., 2020; Liu et al., 2021), which require training on a large number of document-summary pairs. Unfortunately, opinion summarization generally lacks the training pairs with reviews as input and summary as output, as it is difficult and costly for annotators to write summaries for multiple reviews (known as *multi-reviews* in this paper) on a large scale.

Kenny Q. Zhu is the corresponding author, and is partially supported by SJTU-CMBCC Joint Research Scheme and SJTU-Meituan Joint Research Scheme.

Synthetic Summary (sampled from the reviews)		
very **disappointed** in **food** and **service** . the **beef** was **burned** . *the **staff dismissed** my comments* .		
Synthetic Input (to be summarized)		
Textual (unstructured)		
Random	R_1	very average spanish **food** .
	R_2	definitely not a peruvian restaurant .
	R_3	love this place! location is great .
Similarity	R_1	bad **food** and **service** . **disappointed**
	R_2	the **food** and **service** was not good . very **bad experience** .
	R_3	awful **food** and **service** . the **staff** was unfriendly .
Structured		
OpiDig	OA	**disappointed**, **food; disappointed**, **service; burned**, **beef**
Mix-structured		
Ours	OA	not good, **food;** great, location**;** bad, **food; disappointed**, **service;** not fresh, **beef;** unfriendly, **staff;** awful, **service**
	IS	not recommended .**;** my questions were **dismissed** .**;** the **staff** ignored our **comments** .

Table 1: A summary and various inputs synthesized by different methods. Matched words in both input and output are bolded. R denotes textual review. OA indicates explicit opinion-aspect pairs. The italicized sentence doesn't contain explicit OAs, which is called "implicit sentence" (IS). ";" delimits OAs and ISs.

In view of the above challenge, some approaches (Chu and Liu, 2019; Brazinskas et al., 2020b; Isonuma et al., 2021) adopt unsupervised learning, e.g., by using the auto-encoders. They reconstruct individual reviews by encoding themselves at training time, which may prevent them from effectively generating summary by encoding multi-review at test time. Other more popular approaches (Amplayo and Lapata, 2020; Brazinskas et al., 2020a; Amplayo et al., 2021b; Wang and Wan, 2021) focus on creating synthetic (multi-review, summary) pairs for training. They typically sample one or more reviews from all reviews about an entity as the "pseudo" summary, that is, the output of the synthetic training pair. These approaches differ from each other by how the input is created. Existing approaches use either *textual* or *structured* information (see Table 1) to create the input.

The *textual* input is usually a set of reviews sampled from all reviews in the corpus. The most straight-forward method is *leave-one-out* (Brazin-

skas et al., 2020b; Wang and Wan, 2021), which takes all reviews as the input, except for one sampled summary. This results in very long and almost identical inputs, which are not only memory-consuming, but also less effective. Another way is to sample (or synthesize) a subset of the reviews, either randomly (Brazinskas et al., 2020a), or by the similarity between the input text and the summary (Amplayo and Lapata, 2020; Amplayo et al., 2021b). As shown in Table 1, these methods also face challenges because: i) the content of summary and its input may be completely unrelated, such as in *Random* (Brazinskas et al., 2020a); ii) certain important information in the summary can be missing from the input, such as the "beef" in the summary of *Similarity* (Amplayo et al., 2021b). The latter situation is due to some atypical information mentioned in sampled summary that is hardly mentioned in other reviews. Such semantic misalignment between the input and output causes problems when training a summarization model.

Previous works have acknowledged that the most critical information in opinion summarization are the aspects of the product or service and the opinions on these aspects (Angelidis and Lapata, 2018; Mukherjee et al., 2020), such as *beef* $\rightarrow$ *burned*, or *service* $\rightarrow$ *disappointed*. For example, OpiDig (Suhara et al., 2020) uses a review as output while extracting opinion-aspect (OA) pairs from this review as structured input. However, some sentences may not produce any OA pairs at all, such as "*staff dismissed my comments*" in Table 1. Information in these sentences is ignored in previous works. As reviews are usually short and informative, we believe that every sentence in them provides some useful information and should not be ignored. In this paper, we not only make use of the structured opinion-aspect data but also those sentences that do not produce explicit OA pairs. We call the latter implicit sentences (IS). We use a mix of structured and unstructured information as the input of our synthetic training data. Since in real-world scenarios, any input multi-review may contain redundancies and even contradictions that must be properly handled by the summarizer, we further sample OAs and ISs from all other reviews according to sampled summary as mix-structured input. As shown in Table 1, the mix-structured input can cover more information in output than all previous methods.

In order to capture explicit and implicit opinions at the same time, we propose a summarization model with OA encoder and IS encoder. We first pretrain a single encoder by taking only sampled OAs as input, which learns to select important explicit opinions. Then, we fine-tune that model with OA and IS encoder based on pretrained encoder. This allows the information of explicit and implicit opinions to complement each other, leading to reduced information loss in generated summaries.

In summary, this paper makes the following contributions:

1. In preliminary study, we directly convert textual or structured input in previous synthetic datasets to mix-structured input by extracting OAs and ISs from previous input, and discover that mix-structured input is more effective at capturing opinions (Section 3.6).

2. We propose a new data creation method to construct mix-structured synthetic training pair by first sampling a review as summary and then sampling OAs and ISs from other reviews as input. We also design a summarization model with a dual encoder for mix-structured data (Section 2).

3. Compared with previous methods, the proposed model trained on our synthetic data outperforms the state-of-the-art methods on the Yelp, Amazon and RottenTomatos datasets (Section 3.5, Section 3.6, and Section 3.7).

2 Approach

Since there are no reference summaries for training, we design a weakly supervised approach, which first synthesizes the mix-structured training data and then train a dual-encoder summarization model on such data.

2.1 Mix-structured Training Data Creation

Let R denote the set of all reviews about an entity. To avoid information loss in textual inputs or structured inputs, we create a synthetic mix-structured training dataset with four steps, as shown in Figure 1: (1) **extract opinion-aspect pairs (OAs) and implicit sentences (ISs)** from each review in R; (2) **sample a review** as a summary s from R and take the remaining reviews as candidate reviews R_s; (3) **sample OAs** from R_s which describe aspects that are either in or not in s; and (4) **sample ISs** from R_s by similarity with the ISs in s. The

sampled OAs and ISs simulate the OAs and ISs in the actual multi-review to be summarized. We then repeat steps (2)-(4) to create many input-summary training pairs.

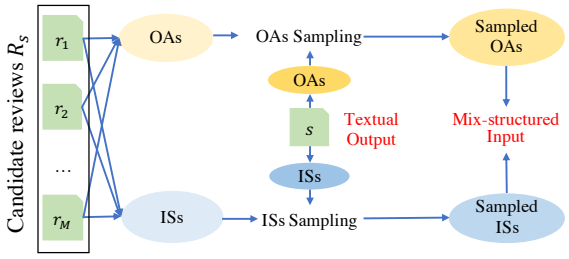


Figure 1: The flow of creating mix-structured data.

2.1.1 OA and IS Extraction

We extract opinion-aspect pairs following the method outlined in (Bhutani et al., 2020a) [1], which first utilizes MIN-MINER (Bhutani et al., 2020b) to parse sentences into dependency trees and then uses a set of syntactic rules (Moghaddam, 2013) to extract OAs [2]. The opinions thus extracted are typically adjectives which either express the reviewer's views or sentiments (e.g., "disappointed") or describe the aspect (e.g., "burned"). Meanwhile, we collect those sentences from which no OAs were extracted, and call these sentences ISs.

2.1.2 Summary Sampling

Given all reviews of an entity, we first randomly sample a review. Then, we extract aspects (A) from the sampled review and extract aspects (A') from other reviews. If A is the subset of A', the sampled review can be seen as a summary. This ensure the sampled review contains only aspects that can be found in other reviews. Because a summary should not discuss things outside of its input. Besides, we ignore those reviews containing first-person singular pronouns and non-alphanumeric symbols except for punctuations (Amplayo and Lapata, 2020), since such reviews don't look like real summaries.

2.1.3 Opinion-Aspect Pairs Sampling

To simulate the actual multi-review, which includes not only all the aspects in the summary, but also some aspects not in the summary, we include two types of OAs in the mix-structured data: *popular* pairs and *unpopular* pairs. The OAs whose aspects are included in the summary are popular, and the other OAs are unpopular. For example, suppose that (*disappointed*, *food*) is an OA in the summary, (*bad*, *food*) is a popular pair and (*not fresh*, *beef*) is an unpopular pair.

For a given (o, a) pair in the summary, we sample a popular pair (o', a) from R_s with a probability which is proportional to the semantic similarity between o and o'. The similarity is computed by the cosine similarity between the average word vectors (Pennington et al., 2014) from o and o'. This is to allow opinions which are different from the mainstream ones to be included in the training data with a small probability. Further, we randomly sample an unpopular pair whose aspect doesn't appear in the summary at all. Such a pair represents an aspect in the input which is ignored by the summary. The number of popular and unpopular pairs to be sampled for each summary is not fixed but follows a normal distribution, to simulate the variations of the number of OAs that are included in different summarization inputs. The determination of parameters of these normal distributions is important and is discussed in Section 2.1.5.

2.1.4 Implicit Sentences Sampling

For a summary s, we sample ISs from R_s with a probability proportional to its similarity with the ISs in s. The similarity is computed by ROUGE-1 recall score. The total number of such IS sampled is also determined by a normal distribution determined below.

2.1.5 Estimating Sample Sizes

We assume the numbers of popular/unpopular OA pairs and ISs that need to be sampled all follow normal distribution, but with different parameters μ and σ. We assume that our summarization model is supposed to produce a summary for N reviews, where N is a parameter of the problem. After removing all the eligible output summaries (each denoted as s) from R, we call the remaining reviews candidate pool R_C. We random sample N [3] reviews at a time from R_C, and regard those (o, a) pairs in which a appears in more than one review in R_C as popular pairs, and all other pairs as unpopular pairs. Note that we are only approximating the popular and unpopular pairs here since we do not know the summary s at this time. We also count

[1] Our framework is not specific to any opinion-aspect extraction algorithm or tool. We choose to use (Bhutani et al., 2020a) because it is an unsupervised method, which is better at synthesizing datasets for cross-domain review corpora (i.e., product, service or movie).

[2] https://github.com/sampoauthors/Sampo

[3] N depends on the number of input reviews of inference sets.

the number of ISs in R_C. After repeating the above random sampling a number of times, we can estimate the μ and σ for the numbers of popular pairs, unpopular pairs and implicit sentences.

2.2 Summarization Models

Given OAs and ISs as mixed input, we design a straight-forward seq2seq model with a dual encoder that deals with the two types of input separately. Next, we present the basic model which is based on the transformer, and then show a way to optimize the model by using an additional pretrain stage.

2.2.1 Basic Model

Our basic model (Figure 2) deals with OAs and ISs in parallel via two transformer encoders, ***OA encoder*** and ***IS encoder***. The parameters of the dual encoder are not shared.

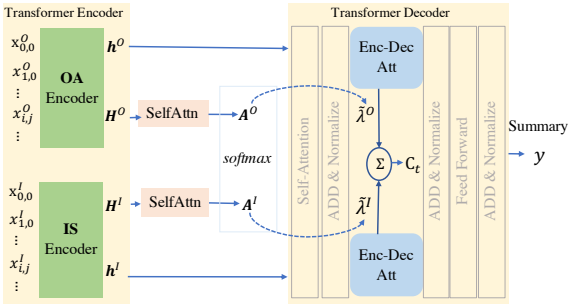


Figure 2: Basic model.

In this model, we concatenate the opinion and aspect in each OA, and arrange all the OAs in a shuffled order. To model the relation among OAs, we add a special token at the beginning of each OA, which indicates the OA representation. $x^O_{i,j}$ is the i^{th} token of j^{th} pair in sampled OAs. The special token of j^{th} pair is $x^O_{0,j}$. The sentences in sampled ISs are concatenated in the same way as sampled OAs. $x^I_{0,j}$ is the special token of j^{th} sentence. We take OAs and ISs as input and summary **y** as output.

Inspired by Chen and Yang (2020), we use H^O_j to denote the encoder state of $x^O_{0,j}$, which indicates the j^{th} pair in sampled OAs. $\mathbf{H}^O = \{H^O_1, ..., H^O_n\}$ consists of all special tokens of OAs. Then, we aggregate the information of all sampled OAs through self-attention mechanism (*SelfAttn*) (Vaswani et al., 2017) and get the OA representation as $\mathbf{A}^O$. Similarly, we can get the IS representation $\mathbf{A}^I$. The *OA probability* $\widetilde{\lambda}^O$ is computed as:

$$\lambda = \frac{\exp(\mathbf{A}^{O\top} v^O)}{\exp(\mathbf{A}^{O\top} v^O) + \exp(\mathbf{A}^{I\top} v^I)} \quad (1)$$

where v^O and v^I respectively denote the randomly initialized context vector of OA encoder and IS encoder. We achieve $\widetilde{\lambda}^O$ by applying temperature on λ for sharpening the attention distribution between encoders, and get *IS probability* $\widetilde{\lambda}^I$ by $1 - \widetilde{\lambda}^O$.

At each decoding step t, C^O_t is the weighted sum of encoder states of OA with *Enc-Dec Attn* as weight. C^I_t is the weighted sum of IS encoder. The combinational context vector C_t for decoding is:

$$C_t = \widetilde{\lambda}^O C^O_t + \widetilde{\lambda}^I C^I_t \quad (2)$$

After inputting C_t to the feed-forward network layer of transformer. The probability distribution of generating y_t is computed by the softmax on C_t.

2.2.2 Optimized Model

In opinion summarization, users' opinions on the aspects of an entity are very important. For optimization, we pretrain a single encoder by setting $\widetilde{\lambda}^I$ as zero and taking only sampled OAs as input and corresponding summary as output. Then, we fine-tune the basic model in Figure 2 by initializing the OA encoder using the parameters of pretrained encoder, enhancing the generated summary with implicit opinion from sampled ISs.

3 Evaluation

We first give the experimental setup, then compare and contrast the end-to-end results against various strong baselines on three datasets, before analyzing the main advantages of our approach.

3.1 Datasets

In this experiment, we use three datasets.

Yelp [4] contains a large number of reviews about consumer services. Each sample in its development and test set (Chu and Liu, 2019) contains 8 reviews and one corresponding human-written summary.

Amazon is a dataset (He and McAuley, 2016) made up of product reviews. Each sample in its development set and test set has 8 reviews and 3 human-written summaries (Brazinskas et al., 2020b).

RottenTomatoes (RT) (Wang and Ling, 2016) is a large set of movie reviews. Each set of reviews (on average 100) about a movie has a human-written gold summary.

We construct the synthetic mix-structured training set for these three datasets. Human-annotated

[4] https://www.yelp.com/dataset

multi-review and summary pairs are used as development and test set (See Table 2 for detail). [5]

	Training	Development	Testing
Yelp	100k	100	100
Amazon	90k	28×3	32×3
RT	25k	536	737

Table 2: Dataset statistics. Training column shows the number of our synthetic training pairs. $\times 3$ means 3 manual summaries per multi-review.

3.2 Implementation Details

When creating synthetic training data, we set N (Section 2.1.5) to be 8 for Yelp and Amazon, and 100 for RT, because it depends on the number of input reviews in each test pair. Each test pair in Yelp or Amazon has 8 reviews as input. Each testing pair in RT has about 100 (on average) reviews as input. During training, we take synthetic OAs and ISs as mix-structured input and sampled summaries as output. At test time, we extract OAs and ISs from multi-reviews in human-annotated test sets as mix-structured input and human-annotated summaries as output. For transformer (Vaswani et al., 2017), we use SGD as the optimizer. The initial learning rate is 0.1, momentum $\beta = 0.1$, decay $\gamma = 0.1$. For BART, we use *bart.large* model with its default settings and fine-tuning it Lewis et al. (2020) with $lr = 3e\text{-}05$. The reason for choosing BART as our basic model is that BART is effective on summarization and easy to use. As our proposed approach is orthogonal to the seq2seq model, we expect our approach to benefit opinion summarization methods regardless of the seq2seq model architecture. We train our models [6] on one RTX 2080Ti GPU with 11G RAM. The average training time of our approaches is about 10 hours.

3.3 Models under Comparison

Different methods trained on their own synthetic training data are listed in Table 3.

Our proposed models can be based on non-pretrained Transformer (Vaswani et al., 2017) and pretrained BART (Lewis et al., 2020). By default, our proposed **basic model (OURS$_{basic}$)** and **optimized model (OURS)** are based on BART.

[5]The size of our synthetic training set does not exceed that of previous synthetic training set.

[6]Data and code: https://github.com/YizhuLiu/Opinion-Summarization

MeanSum (Chu and Liu, 2019): An autoencoder model, which decodes the summary based on the mean representation of input reviews.
Copycat (Brazinskas et al., 2020b): A hierarchical variational autoencoder model with the controlling of novelty.
OpiDig (Suhara et al., 2020): A self-training model with a review as output and the OAs of this review as input for training.
Denoise (Amplayo and Lapata, 2020): A denoising summarization model with linguistically motivated noising datasets.
FewSum (Brazinskas et al., 2020a): A conditional transformer model, including content coverage, writing style and length deviations.
PlanSum (Amplayo et al., 2021b): A summarization model trained on sentiment and aspect distributions of reviews.
TranSum (Wang and Wan, 2021): A summarization model with sentiment and aspect embeddings as input.

Table 3: Different opinion summarization approaches.

3.4 Evaluation Metrics

We show the *automatic metrics* and *human evaluation* [7] below.

ROUGE (Lin, 2004) (F1) include ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L). [8]

Diversity (Div↓) uses self-BLEU (Zhu et al., 2018) [9] which measures BLEU scores of each generated sentence by considering others as reference. The lower value means more diversity.

Aspect Coverage (AC) measures the overlapping of aspects in the gold summary and generated summary. We ask 3 annotators to extract aspects from text. Given a generated summary and its gold summary, annotators manually extract aspects from them. We take aspects of gold summary as reference, and compute R-1 recall between the reference and aspects of generated summaries.

Human Evaluation. We randomly select 50, 32 (all test data) and 50 samples from the test sets of Yelp, Amazon and RT, respectively. We ask three human annotators, who are native or proficient English speakers to rank gold summary and summaries generated by OURS and TranSum under 5 aspects: *Fluency* (Flu.), *Non-redundancy* (NR.), *Opinion Consistency* (Cons.) and *Overall*. Kiritchenko and Mohammad (2016) show that Best-Worst Scaling (Louviere et al., 2015) is more reliable. Thus, we apply Best-Worst Scaling on individual aspects, which computes the percentage of times a model was selected as the best minus the percentage of times it was selected as the worst.

[7]The Fleiss' Kappa coefficient of judges is 0.63, indicating substantial agreement.

[8]Previous methods used different toolkits, so the ROUGE scores in some published papers cannot be compared. To be fair, we use *pyrouge* (https://pypi.org/project/pyrouge). We re-evaluate Copycat and FewSum by pyrouge, so their results may be slightly different from their published versions.

[9]https://github.com/geek-ai/Texygen

Approach	Yelp					Amazon					RT				
	R-1	R-2	R-L	AC	Div↓	R-1	R-2	R-L	AC	Div↓	R-1	R-2	R-L	AC	Div↓
Meansum	28.86	3.66	15.19	0.34	0.38	29.20	4.70	18.15	0.17	0.40	15.79	1.91	12.26	0.13	0.28
Copycat	29.47	5.26	18.09	0.38	0.34	31.97	5.81	20.16	0.18	0.43	14.98	3.07	12.19	0.13	0.28
OpiDig	29.96	5.00	17.33	0.39	0.33	29.02	5.14	17.73	0.23	0.32	14.21	1.82	10.23	0.15	0.27
Denoise	30.14	4.99	17.65	0.39	0.27	31.76	5.85	19.87	0.22	0.27	21.26	4.61	16.27	0.16	0.25
FewSum	31.96	5.64	17.77	0.38	0.28	32.04	5.93	20.03	0.20	0.30	20.44	4.79	16.12	0.15	0.26
OURS w/o PLM	33.43	6.27	18.37	0.41	0.22	32.57	6.19	20.18	0.33	0.26	21.56	5.23	17.00	0.17	0.24
PlanSum	34.79	7.01	19.74	0.40	0.26	32.87	6.12	19.05	0.23	0.32	21.77	6.18	16.98	0.16	0.24
TranSum	36.62	8.41	20.31	0.38	0.27	34.23	7.24	20.49	0.23	0.32	25.34	8.62	18.35	0.16	0.25
OURS	**36.78**	**8.66**	**20.52**	**0.44**	**0.20**	**34.50**	**7.64**	**20.73**	**0.34**	**0.25**	**25.44**	**8.75**	**18.52**	**0.17**	**0.23**

Table 4: Automatic evaluation. The scores underlined are significantly better than TranSum ($p<0.05$). "PLM" means pretrained language model.

3.5 Main Results

As shown in Table 4 , among all the models that don't use any pretrained language model, OURS w/o PLM performs the best. Compared with TranSum and PlanSum, the ROUGE scores of OURS is lower because TranSum and PlanSum introduce external knowledge by using pretrained BERT (Devlin et al., 2019). OURS using pretrained BART achieves the best scores in terms of ROUGE, AC and Div↓, demonstrating the full strength of our created synthetic training data. In general, our approaches receive higher AC scores but lower Div scores since the model benefits from directly taking OAs as input and captures complementary information from ISs. The summary generated by OURS in Table 5 contains more implicit information, such as "patio seating". Compared to Amazon, the advantage of our best model on Yelp is less because each sample has 3 gold summaries in Amazon and has only one gold summary in Yelp. The improved generated summaries are more likely to match tokens in multiple gold summaries. Thus, even if the generated summary on yelp covers more important information, the rouge score may not change much.

Table 5 compares the summaries generated by the best models trained on structured data (OpiDig), textual data (TranSum) and mix-structured data (OURS). OpiDig clusters OAs of multi-review and then uses the center pairs to construct summaries, which may be weakened by the noise of clustering. As shown in Table 5, "atmosphere" and "portions" of OpiDig's summary are not in gold summary. Moreover, it is difficult to generate summary sentences without OAs, such as the italicized sentences in gold summary. Compared with OpiDig, OURS trained on mix-structured data learns to expand important OAs as sentences with explicit opinions and summarize ISs for implicit opinions, which can cover more gold aspects and generate similar implicit sentences to gold summary. For TranSum, as certain information in sampled summary cannot be found from the synthetic textual inputs during training, the summaries generated by TranSum may lose important information and contain redundant information, such as "margaritas" in gold summary and the italicized sentence about "price". OURS avoids the above problems because of the fine-grained synthetic input for the sampled summary. Since the cost of writing opinion summaries for multi-reviews is very high, the size of test sets is small. The improvement of our proposed approach on all three different testing sets shows that our results are reliable.

Gold	the **servers** are kind and knowledgeable . *they will patiently answer your questions . they offer* ***patio seating*** . the free **chips** and **salsa** are always a plus , and the **margaritas** are **amazing** too . the menu is full tasty **authentic mexican food.**
OpiDig	a pretty **place** . the **service** is amazing and the **food** is amazing . **atmosphere** is great and the **portions** are huge
TranSum	i love this **place** . the **food** is good and the **service** is great . the **chips** and **salsa platter** is huge enough. *the only thing is that it 's a little pricey for what you get .*
OURS	it 's one of the **authentic mexican restaurant** in the area. the **food** is great . the **servers** are very friendly and knowledgeable . *they took the order patiently* . the **chips** and **salsa** are good too . *it is huge and has* ***patio seating*** .

Table 5: Generated summaries and their gold summary from Yelp. Bolded words are aspects. The sentences in red don't match Gold summary. The italicized sentences are ISs.

In human evaluation, we assess summaries generated by the previous state-of-the-art model (TranSum), our best model (OURS) and gold summaries. Gold gets best manual score in Table 6 since the gold summaries are written by human. The summaries generated by our model are better than TranSum from all perspectives.

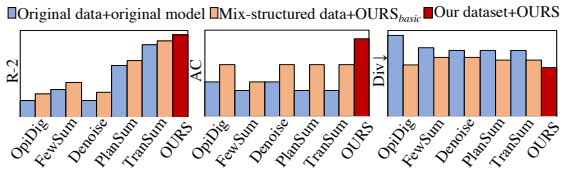
3.6 Mix-structured vs. Structured or Unstructured Synthetic Data

To evaluate the effectiveness of mix-structured data versus other kinds of synthetic data, we convert structured and non-structured synthetic input into mix-structured version, and compare previous approaches trained on their original synthetic data

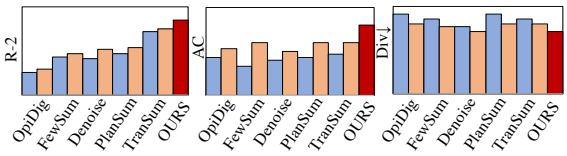
Data	Model	Flu	Coh	NR	Cons	Overall
Yelp	Gold	0.34	0.49	0.41	0.35	0.31
	TranSum	-0.46	-0.53	-0.70	-0.64	-0.48
	OURS	0.12	0.14	0.29	0.29	0.17
Amazon	Gold	0.32	0.55	0.38	0.44	0.32
	TranSum	-0.54	-0.67	-0.68	-0.72	-0.41
	OURS	0.22	0.12	0.30	0.28	0.09
RT	Gold	0.42	0.36	0.51	0.29	0.45
	TranSum	-0.61	-0.64	-0.66	-0.46	-0.48
	OURS	0.19	0.28	0.15	0.17	0.05

Table 6: Human evaluation.

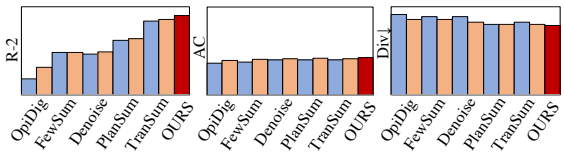
and OURS$_{basic}$ trained on the mix-structured version of previous synthetic data. This is the best way to compare data in different types, as different synthetic datasets must be accompanied by their compatible models. Results are shown in Figure 3.



(a) Yelp



(b) Amazon



(c) RT

Figure 3: Results on different synthetic data and their applicable models. Datasets in mix-structured version ("mix-") use OURS model for training.

For datasets with textual multi-review as input (FewSum, Denoise, PlanSum and TranSum), we convert them into mix-structured versions by extracting OAs and ISs from their multi-reviews as input. For synthetic data with structured input (OpiDig), we construct its mix-structured version by sampling OAs and ISs for each output in original dataset through our data creation method and taking them as input. To be fair, for OpiDig, FewSum and Denoise, the training on their original textual datasets did not use any pretrained language model, so we train OURS without pretrained language models on their datasets in mix-structured version.

For mix-structured version data of PlanSum and TranSum, we train OURS with pretrained language model on them, as PlanSam and TranSum trained on original textual data used pretrained language model to import external knowledge. In Figure 3, OURS$_{basic}$ trained on the mix-structured version of previous textual synthetic datasets is better than previous approaches in terms of R-2, AC and Div↓ scores, showing that mix-structured data is helpful in highlighting the aspects and opinions. Compared with the structured datasets OpiDig, the mix-OpiDig get better ROUGE scores and Div↓ scores as the ISs in mix-structured data help the model capture implicit information.

As shown in Figure 3, OURS trained on our dataset performs best since we sample OAs and ISs from all reviews except summaries and use optimized model for training, which makes the best use of mix-structured data.

To explain the rationality of mix-structured data, we call the sentences that do include OAs as *explicit sentences* (ES). We remove OAs and stopwords from ESs and compute the percentage of the remaining tokens in original ESs. These percentages are 10.7% in Yelp, 11.1% in Amazon, and 4.3% in RT. We also randomly sample 100 reviews from each dataset and ask human annotators to pick the sentences that still contain useful information after removing OAs. Result shows that $< 10\%$ of ESs contain useful residual information, i.e., 9.1% (Yelp), 9.3% (Amazon), 4.0% (RT). The above results show that ESs contain very little implicit information. Therefore, we take extract OAs from reviews as part of the input is reasonable. Meanwhile, we calculate the distribution of ESs and ISs in multi-reviews and reference summaries in human-annotated test sets. As shown in Table 7, the proportion of ISs in multi-review and reference summary is more than 30%, which means that adding ISs is important and necessary.

Dataset	Multi-review		Reference summary	
	ESs	ISs	ESs	ISs
Yelp	0.54	0.46	0.68	0.32
Amazon	0.55	0.45	0.63	0.37
RT	0.33	0.67	0.32	0.68

Table 7: Proportion of ES and IS in test sets.

3.7 Ablation

We use various ablation studies on our synthetic dataset and proposed models. We report R-2, AC and Div↓ scores on test sets.

3.7.1 Mix-structured synthetic data

In this section, we justify our method of sampling OAs and ISs and estimating their sampling sizes.

OA & IS sampling	Sampling sizes	Yelp	Amazon	RT
Similarity	Distribution	**8.66/0.44/0.20**	**7.64/0.34/0.25**	**8.75/0.17/0.23**
Random	Distribution	6.73/0.23/0.22	3.48/0.22/0.25	6.44/0.14/0.24
Similarity	Average	7.43/0.42/0.20	6.92/0.30/0.26	7.90/0.17/0.24

Table 8: Results of OURS trained on data created by different ways and different sample sizes.

Sampling OAs and ISs. To show the effectiveness of our sampling method based on similarity distribution, we randomly sample OAs and ISs, and estimate sampling sizes following Section 2.1.5. Table 8 shows that the sampling based on similarity is better than random sampling in terms of ROUGE, AC and Div, indicating that our sampling way is more similar to real distribution of OA and IS in multi-reviews.

Sampling sizes. Table 8 compares the samplings based on same similarity distribution (Section 2.1.3 and Section 2.1.4) but different sampling sizes. As a baseline, unlike our sampling sizes estimation based on a normal distribution, we compute the average (n) of the numbers of OAs in N reviews, and take $\frac{n}{2}$ as sampling size of popular and unpopular pairs respectively. In this way, the total number of synthesized OAs will be similar to the total number of OAs in multiple reviews. We also take the average of the numbers of ISs in N reviews as sampling size of implicit sentences. Table 8 shows that our proposed method for estimating sampling sizes is better, since it better simulates the number of OAs and ISs in real world scenarios.

3.7.2 Our proposed models

We compare different models designed for mix-structured data in Table 9. OURS performs better on all evaluation metrics, meaning the summaries from OURS can cover more important aspects and generate more accurate implicit sentences.

Model		Yelp	Amazon	RT
w/o PLM	OURS$_{basic}$	5.55/0.33/0.24	5.73/0.24/0.27	4.92/0.14/0.25
	OURS	**6.07/0.41/0.22**	**6.19/0.33/0.26**	**5.23/0.17/0.24**
with PLM	OURS$_{basic}$	7.94/0.38/0.22	7.23/0.28/0.26	8.63/0.16/0.23
	OURS	**8.66/0.44/0.20**	**7.64/0.34/0.25**	**8.75/0.17/0.23**

Table 9: R-2/AC/Div↓ of our generated summaries.

To make full use of mix-structured data, we first train a seq2seq model with a single encoder taking sampled OAs as input, and then fine-tuned our basic model with a dual encoder taking OAs and ISs as input. The pretraining of the single encoder taking only OAs as input enables the model better to select OAs. Compared with OURS$_{basic}$ summary in Table 10, OURS summary in Table 5 can capture the opinions and aspects more accurately due to the extra OA pretraining phase and the last sentence of OURS summary is inferred by adding IS encoder.

However, compared with gold summary, the sentence in generated summaries of OURS are not coherent enough and the coreference is not clear. As shown in Table 5, the sentences in OURS output on food are incoherent and the 'it' in the last sentence denotes the restaurant. The reason is that some reviews in corpus are abbreviated or non-standard, which brings noise to the datasets and models.

OURS$_{basic}$	**great food** , **great service** , **great atmosphere** , and **great prices** . i have been there a few times and have **never** had a **bad experience** .

Table 10: Summary generated by OURS$_{basic}$ on the same multi-review as Table 5.

4 Related Work

Opinion summarization has a special focus on aspects of the product or service, making it different from other multi-document tasks, such as news summarization (Fabbri et al., 2019).

Opinion summarization suffers from a lack of training pairs. Some work (Chu and Liu, 2019; Brazinskas et al., 2020b; Isonuma et al., 2021) used autoencoder to train the model by reconstructing loss or sentence embeddings. Others create synthetic datasets for supervised training. The input format of synthetic datasets is textual or structured. For the textual input, some approaches (Brazinskas et al., 2020a; Wang and Wan, 2021) regarded one review as a summary and took all or part of the rest as input. Wang and Wan (2021) computed the distance between the summary and all remaining reviews as weights of review embeddings. Amplayo et al. (2021b) took the nearest neighbors as inputs based on review representations. Amplayo and Lapata (2020) added noise to the sampled summary from the segment noising and document noising by replacing the whole review with a similar one. Elsahar et al. (2021) labeled input reviews and sampled summaries with control tokens and took control tokens as prefixes at decoding. However, these datasets is limited by biased reviews, which cannot be summarized from other reviews.

The aspects (Luo et al., 2018, 2019) and opin-

ions are quite important for opinion summarization. Some approaches (Angelidis and Lapata, 2018; Mukherjee et al., 2020) classified the sentences of reviews into different aspects and collected the most salient sentence of each class as summary. Tian et al. (2019) classified words into three types (i.e., aspect, opinion and context) and predicted summary by the probability distribution on these types. Inspired by these works, Suhara et al. (2020) extracted opinion-aspect phrases from each review and transformed the task into single document summarization. Amplayo et al. (2021a) used predefined aspects to construct synthetic training data and trained a controllable model to generate summaries based on aspects. However, all of these works neglect some other information in the sentences which cannot be explicitly formulated as opinion-aspect pairs.

Thus, we create a mix-structured synthetic dataset consisting of opinion-aspect pairs and implicit sentences, which can get more accurate and comprehensive summaries.

5 Conclusion

In this work, we proposed a new method to generate mix-structured synthetic training data for opinion summarization. We designed a transformer-based seq2seq model with a dual encoder to deal with OAs and ISs separately. The results showed that our approach can make full use of mix-structured data and generate better opinion summaries.

6 Limitations

The limitation of our proposed approach is that it is more sensitive to explicit opinion information and more effective in reviews with explicit opinions.

We observe that the performance of the models on different datasets. Table 9 shows that OURS trained on RT improves most because the movie reviews in RT contain more ISs, such as character and plot descriptions after using ISs. OURS performs better than OURS$_{basic}$ because of the pretraining on single encoder with only sampled OAs as input. The difference between OURS and OURS$_{basic}$ trained on Yelp and Amazon is greater than that trained on RT, because Yelp and Amazon contain more OAs than RT. Thus, our approach is more effective on the reviews with more explicit opinions, like Yelp and Amazon.

Table 4 shows that although our model achieves the best among all for RT, the margin here becomes even less. This is because the summaries in RT are shorter with fewer explicit OAs. Movie reviews include the discussion on plots, such as italicized sentences of Gold in Table 11, which makes the proportion of OAs in movie reviews less than that of Yelp and Amazon. As shown in Table 11, even though the summary generated by OURS represents more information of gold summary than other baselines, it is not much closer to gold summary. Therefore, the gain of our proposed OURS on RT is less than the other two datasets.

Gold	***movie*** *begins with promise* . but it suffers from a flimsy **narrative** and poor **execution** . *with alien-sized plot holes*
OpiDig	great **concept** . a strange but cool **comedy** .
TranSum	**hancock** is a **movie** that's a lot of fun, but it'll be a bit of the same time as the **movie**.
OURS	**hancock** has a promising premise , but the **narrative** slips into a confusing **backstory** .

Table 11: Summaries generated by different models and their gold summary (Gold) from RT. Bolded words are aspects. The sentences in red don't match gold summary. The italicized sentences are ISs.

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021a. Aspect-controllable opinion summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6578–6593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021b. Unsupervised opinion summarization with content planning. In *AAAI*.

Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised opinion summarization with noising and denoising. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics.

Nikita Bhutani, Aaron Traylor, Chen Chen, Xiaolan Wang, Behzad Golshan, and Wang-Chiew Tan. 2020a. Sampo: Unsupervised knowledge base construction for opinions and implications. In *Automated Knowledge Base Construction*.

Nikita Bhutani, Aaron Traylor, Chen Chen, Xiaolan Wang, Behzad Golshan, and Wang-Chiew Tan. 2020b. Sampo: Unsupervised knowledge base construction for opinions and implications. In *Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, June 22-24, 2020*.

Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2020a. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics.

Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2020b. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics.

Eric Chu and Peter J. Liu. 2019. Meansum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

Hady Elsahar, Maximin Coavoux, Jos Rozen, and Matthias Gallé. 2021. Self-supervised and controlled multi-document opinion summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*. Association for Computational Linguistics.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. WWW '16. International World Wide Web Conferences Steering Committee.

Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2021. Unsupervised abstractive opinion summarization by generating sentences with tree-structured topic guidance. *arXiv preprint arXiv:2106.08007*.

Svetlana Kiritchenko and Saif M. Mohammad. 2016. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018a. Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yizhu Liu, Qi Jia, and Kenny Zhu. 2021. Keyword-aware abstractive summarization by extracting set-level intermediate summaries. In *Proceedings of the Web Conference 2021*, pages 3042–3054.

Yizhu Liu, Zhiyi Luo, and Kenny Q. Zhu. 2018b. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Jordan J Louviere, Terry N Flynn, and Anthony Alfred John Marley. 2015. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.

Zhiyi Luo, Shanshan Huang, Frank F Xu, Bill Yuchen Lin, Hanyuan Shi, and Kenny Q. Zhu. 2018. Extra: Extracting prominent review aspects from customer feedback. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3477–3486.

Zhiyi Luo, Shanshan Huang, and Kenny Q. Zhu. 2019. Knowledge empowered prominent aspect extraction from product reviews. *Information Processing & Management*, 56(3):408–423.

Congbo Ma. 2021. Improving deep learning based multi-document summarization through linguistic knowledge. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, page 2704. ACM.

Samaneh Abbasi Moghaddam. 2013. Aspect-based opinion mining in online reviews. In *Applied Sciences: School of Computing Science*.

Rajdeep Mukherjee, Hari Chandana Peruri, Uppada Vishnu, Pawan Goyal, Sourangshu Bhattacharya, and Niloy Ganguly. 2020. Read what you need: Controllable aspect-based opinion summarization of tourist reviews. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM.

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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*.

Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. 2020. Opiniondigest: A simple framework for opinion summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics.

Yufei Tian, Jianfei Yu, and Jing Jiang. 2019. Aspect and opinion aware abstractive review summarization with reinforced hard typed decoder. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*.

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*. Association for Computational Linguistics.

Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 47–57.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM.