Unsupervised Opinion Summarization with Content Planning

Reinald Kim Amplayo, Stefanos Angelidis, Mirella Lapata
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
reinald.kim@ed.ac.uk, s.angelidis@ed.ac.uk, mlap@inf.ed.ac.uk

Abstract

The recent success of deep learning techniques for abstractive summarization is predicated on the availability of large-scale datasets. When summarizing reviews (e.g., for products or movies), such training data is neither available nor can be easily sourced, motivating the development of methods which rely on *synthetic* datasets for supervised training. We show that explicitly incorporating *content planning* in a summarization model not only yields output of higher quality, but also allows the creation of synthetic datasets which are more natural, resembling real world document-summary pairs. Our content plans take the form of aspect and sentiment distributions which we induce from data without access to expensive annotations. Synthetic datasets are created by sampling pseudo-reviews from a Dirichlet distribution parametrized by our content planner, while our model generates summaries based on input reviews and induced content plans. Experimental results on three domains show that our approach outperforms competitive models in generating informative, coherent, and fluent summaries that capture opinion consensus.

Introduction

The large volume of online product reviews has led to the proliferation of automatic methods for digesting their content in order to facilitate decision making. The fields of opinion mining and sentiment analysis (Pang and Lee 2008) have offered various solutions, ranging from sentiment classification (Pang, Lee, and Vaithyanathan 2002), to aspect extraction (Mukherjee and Liu 2012), and aspect-based sentiment analysis (Pontiki et al. 2016). Beyond extracting surface-level information (e.g., sentiment labels from reviews), effective summarization systems (Hu and Liu 2006) are needed to succinctly convey opinions to users, e.g., to condense multiple reviews for a given product and identify which weaknesses and features to pay attention to.

Due to the absence of opinion summaries in review websites and the difficulty of annotating them on a large scale, most previous work has relied on extractive approaches (Ku, Liang, and Chen 2006; Paul, Zhai, and Girju 2010; Carenini, Cheung, and Pauls 2013; Angelidis and Lapata 2018), where parts of the input reviews are copied and arranged onto a summary. More recent methods (Chu and Liu 2019; Amplayo and Lapata 2020; Bražinskas, Lapata, and Titov 2019) focus on generating abstractive summaries which can be more informative and less redundant compared to cut-and-paste extracts. They consider an *unsupervised learning* setting where there are only documents (product or business reviews) available without corresponding summaries. An intuitive solution to the lack of training data is to create synthetic summary-review pairs (Amplayo and Lapata 2020; Bražinskas, Lapata, and Titov 2019) by sampling a review from a corpus of product reviews, and pretending it is a summary.

Input Reviews

1. *Local dive bar experience! Authentic phoenix experience squished behind the starbucks.* Pros: Decent prices, $2 mystery shots, *clean bathroom* ...
2. Cheap drinks, awesome bar staff, *stiff pours* ...
3. Cheap drinks, great happy hour (that's ridiculously long and cheap) ... I've only found great bartenders and patrons at this little bar ...
4. It's a local bar with *no frills except pool table, bar,* and friendly people ... *The sliding glass door with the little beach is what makes this place awesome!!!* ...
5. Bartender was friendly and made great shots, *but the place was full of regulars who made it impossible to have fun* ...
6. *Their Christmas decorations rival that of coach house but without the Scottsdale crowd.* You can find every type of person hanging out here. The staff is friendly ...
7. ... reminds me of back home in the Mid West. Good times and great spot to mingle and meet new people!
8. Lynn is the reason I continue to come back!! She is personable, fun, and dedicated.

Opinion Summary

The drinks here are well priced, especially during happy hour. There is a large variety of regulars from various backgrounds and ages. Great place to meet new people. The staff are great they provide a nice judgement free environment and they aren't stingy on the pours.

Figure 1: Yelp reviews about a local bar and corresponding summary. Aspect-specific opinions are in color (e.g., drinks, guests, staff), while less salient opinions are shown in *italics*.

Although synthetic datasets enable the use of supervised training and have been found to produce higher quality summaries than autoencoder-based methods (Chu and Liu 2019), they cannot, by definition, resemble real-world data. Bražinskas, Lapata, and Titov (2019) rely on random sampling to select the pseudo-summary which might have no connection to the input it purports to summarize. Amplayo and Lapata (2020) create multiple input reviews by adding noise to the sampled summary. They generate syntactically noisy versions or extract lexically similar reviews under the unrealistic assumption that all reviews with overlapping vocabulary will be semantically similar to the summary. As shown in Table 1, real-world reviews discuss a variety of opinions covering different aspects of the entity under consideration (e.g., for a bar it might be the price of the drinks, the stuff, the atmosphere of the place). Some of these aspects are salient, we expect to see them mentioned in the summary and discussed in most reviews, while others will be less salient and absent from the summary. There is also variety among reviews: some will focus on several aspects, others on a single one, and there will be some which will discuss idiosyncratic details.

In this paper, we propose to incorporate *content planning* in unsupervised opinion summarization. The generation literature provides multiple examples of content planning components (Kukich 1983; McKeown 1985) for various domains and tasks including data-to-text generation (Gehrmann et al. 2018; Puduppully, Dong, and Lapata 2019), argument generation (Hua and Wang 2019), and summarization (Kan and McKeown 2002). Aside from guiding generation towards more informative text, we argue that content plans can be usefully employed to reflect a natural variation of sampled reviews in creating a synthetic dataset. Our content plans take the form of aspect and sentiment probability distributions which are induced from data without access to expensive annotations. Using these as parameters to a Dirichlet distribution, we create a synthetic dataset of review-summary pairs, where the variation of aspect mentions among reviews can be controlled. We also propose an opinion summarization model that uses these distributions as a content plan to guide the generation of abstractive summaries.

Experiments on three datasets (Wang and Ling 2016; Chu and Liu 2019; Bražinskas, Lapata, and Titov 2019) representing different domains (movies, business, and product reviews) and summarization requirements (short vs longer summaries) show that our approach outperforms competitive systems in terms of ROUGE, achieving state of the art across the board. Human evaluation further confirms that the summaries produced by our model capture salient opinions as well as being coherent and fluent.

Related Work

Most previous work on unsupervised opinion summarization has focused on extractive approaches (Ku, Liang, and Chen 2006; Paul, Zhai, and Girju 2010; Carenini, Cheung, and Pauls 2013; Angelidis and Lapata 2018) which cluster opinions of the same aspect or sentiment, and identify text that represents each cluster. There have been relatively fewer attempts to create abstractive summaries. Ganesan, Zhai, and Han (2010) generate summaries from textual graphs while other work (Carenini, Ng, and Pauls 2006; Di Fabbrizio, Stent, and Gaizauskas 2014) employs a two-stage framework that first selects salient text units and then generates an abstractive summary based on templates.

The majority of eural summarization models (Rush, Chopra, and Weston 2015; See, Liu, and Manning 2017) make use of the very successful encoder-decoder architecture (Sutskever, Vinyals, and Le 2014), often enhanced with attention (Bahdanau, Cho, and Bengio 2014) and copy mechanisms (Vinyals, Fortunato, and Jaitly 2015) which have been shown to encourage diversity and fluency in the output. Unsupervised text generation methods (Freitag and Roy 2018; Fevry and Phang 2018; Chu and Liu 2019) conventionally make use of variational autoencoders (Kingma and Welling 2014), while employing relatively simple decoders in order to mitigate posterior collapse (Kingma and Welling 2014; Bowman et al. 2016). A more recent line of work (Bražinskas, Lapata, and Titov 2019; Amplayo and Lapata 2020) creates synthetic datasets in cases where gold standard summaries are not available which in turn allow to train models in a supervised setting and make use of of effective decoding techniques such as attention and copy. Our method is in line with this work, but ultimately different in its use of content planning to guide both summarization and synthetic data creation.

Content plans have been successfully used to improve generation performance in both traditional (Kukich 1983; McKeown 1985) and neural-based systems (Gehrmann et al. 2018; Puduppully, Dong, and Lapata 2019). Content plans are often discrete and designed with a specific task and domain in mind. Examples include a sequence of facts for data-to-text generation (Gehrmann et al. 2018; Moryossef, Goldberg, and Dagan 2019), a list of Wikipedia key-phrases for argument generation (Hua and Wang 2019), and entity mentions and their clusters in news summarization (Amplayo, Lim, and Hwang 2018; Sharma et al. 2019). Our content plans are neither discrete nor domain-specific. They take the form of aspect and sentiment distributions, and serve the dual purpose of creating more naturalistic datasets for model training and guiding the decoder towards more informative summaries.

Problem Formulation

We assume access to a collection of reviews about a specific entity (e.g., a movie, product, business). These reviews have ratings, which suggest the overall *sentiment* of the reviews and can be either binary (e.g., positive or negative) or on a scale (e.g., from 1 to 5). We further assume that reviews typically focus on certain *aspects* of the entity, which are features subject to user opinions (e.g., the price and image quality of a television, the acting and plot of a movie). Finally, we do not assume access to gold-standard summaries, since in most domains these do not exist.

Let $\mathbf{X} = \{\mathbf{x}_i\}$ denote the set of reviews about an entity. The goal of opinion summarization is to generate a summary $\mathbf{y}$ that covers salient opinions mentioned in the majority of the reviews. For each review, we first induce aspect and sentiment probability distributions $p(a)$ and $p(s)$. We do this with a content plan induction model which learns to reconstruct the review from aspect and sentiment embeddings. Distributions $p(a)$ and $p(s)$ are then used to create a synthetic dataset $\mathbb{D} = \{\mathbf{X}, \mathbf{y}\}$ of review-summary pairs.

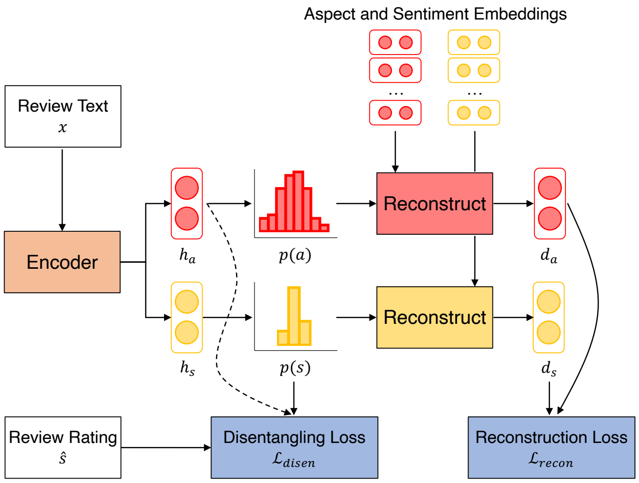

Figure 2: Model architecture of our content plan induction model. The dotted line indicates that a reverse gradient function is applied.

We make use of the Dirichlet distribution parameterized with $p(a)$ and $p(s)$ for sampling, which ensures that the reviews are naturally varied and the summary is representative the opinions found in reviews. Finally, we generate opinion summary $\mathbf{y}$ using a summarization model, which is conditioned on the input reviews $\mathbf{X}$, but also guided by distributions $p(a)$ and $p(s)$, which we view as a content plan.

Content Plan Induction

Our content plan induction model is illustrated in Figure 2. It induces probability distributions $p(a)$ and $p(s)$ from review $\mathbf{x}$ by learning aspect and sentiment embeddings, and reconstructing the encoding of $\mathbf{x}$ through these embeddings. It is similar to neural topic models for aspect extraction (He et al. 2017; Angelidis and Lapata 2018), but also learns sentiment representations.

We encode review $x = \{w_1, ..., w_N\}$ using a neural BiLSTM (Hochreiter and Schmidhuber 1997) followed by a mean pooling operation. The output encoding is split into aspect- and sentiment-specific document encodings, h_a and h_s, respectively, which are used in softmax classifiers to obtain distributions $p(a)$ and $p(s)$ (see Figure 2):

$$\{h_i\} = \text{BiLSTM}(\{w_i\}) \tag{1}$$

$$h_a, h_s = \sum\nolimits_i h_i / N \tag{2}$$

$$p(a) = \text{softmax}(W_a h_a + b_a) \tag{3}$$

$$p(s) = \text{softmax}(W_s h_s + b_s) \tag{4}$$

where N is the number of review tokens, and W_a and W_s are weight matrices.

We learn aspect and sentiment embedding matrices $\mathbf{A}$ and $\mathbf{S}$, via reconstructing the review. We obtain reconstructions d_a and d_s by weight-summing embeddings using $p(a)$ and $p(s)$:

$$d_a = \sum\nolimits_i \mathbf{A}_i * p(a_i) \tag{5}$$

$$d_s = \sum\nolimits_i \mathbf{S}_i * p(s_i) \tag{6}$$

The model is trained using two different objectives. Firstly, a contrastive max-margin objective function is used to reconstruct the original encodings h_a and h_s with d_a and d_s, respectively. For each review $\mathbf{x}$, we randomly sample m reviews as negative samples and obtain encodings $\{n_a^{(i)}, n_s^{(i)}\}$ for $1 \leq i \leq m$. We formulate the objective function as a hinge loss $\mathcal{L}_{recon}$ that maximizes the inner product between d_a and d_s and the original encodings and minimizes the inner product between d_a and d_s and the negative samples. We additionally ensure diversity among aspect/sentiment embeddings in memory (He et al. 2017) by adding a regularization term $\mathcal{R}_{recon}$ to encourage uniqueness:

$$\begin{aligned} \mathcal{L}_{recon} &= \sum\nolimits_i \max(0, 1 - d_a h_a + d_a n_a^{(i)}) \\ &\quad + \sum\nolimits_i \max(0, 1 - d_s h_s + d_s n_s^{(i)}) \end{aligned} \tag{7}$$

$$\mathcal{R}_{recon} = \|\mathbf{A}\mathbf{A}^\top - \mathbf{I}\| + \|\mathbf{S}\mathbf{S}^\top - \mathbf{I}\| \tag{8}$$

where $\mathbf{I}$ is the identity matrix. $\mathcal{R}_{recon}$ minimizes the dot product between two different embeddings in memory, encouraging orthogonality.

We also ensure that the aspect embedding matrix $\mathbf{A}$ does not include information regarding sentiment, and vice versa, by adding a disentanglement loss $\mathcal{L}_{disen}$. This is important since we want to use aspect information to plan the summary content without bias towards a certain sentiment. To distinguish sentiment information, we leverage review ratings $\hat{s}$ as sentiment labels and employ a cross-entropy loss with respect to sentiment distribution $p(s)$. We also predict the same review ratings $\hat{s}$ given aspect-specific document encoding h_a as input. For this, we use an adversarial classifier with a reverse gradient function (Ganin et al. 2016) which reverses the sign of the gradient during backpropagation. This objective learns the opposite of classifying and thus removes sentiment information from aspect embeddings $\mathbf{A}$. We use the following (adversarial) cross-entropy objective as our disentanglement loss:

$$\begin{aligned} p(s)_{adv} &= \text{softmax}(\text{GradRev}(W_{adv} h_a + b_{adv})) \\ \mathcal{L}_{disen} &= -\log p(\hat{s}) - \log p(\hat{s})_{adv} \end{aligned} \tag{9}$$

The overall training loss is the linear addition of the reconstruction and disentanglement losses, and the regularization term mentioned above (λ is a hyperparameter controlling the regularization):

$$\mathcal{L}_{induce} = \mathcal{L}_{recon} + \mathcal{L}_{disen} + \lambda \mathcal{R}_{recon} \tag{10}$$

After training, we obtain probability distributions $p(a)$ and $p(s)$ for each review, and use them to create a synthetic dataset and train a summarization model.

Synthetic Dataset Creation

To create synthetic dataset $\mathbb{D} = \{\mathbf{X}, \mathbf{y}\}$, we first sample a review from the corpus and pretend it is summary $\mathbf{y}$. Next, we sample a set of reviews $\mathbf{X}$ conditioned on $\mathbf{y}$ and pretend they serve as the input which led to summary $\mathbf{y}$. We impose a few (stylistic) constraints on the selection of candidate summaries to ensure that they resemble actual summaries. We discuss these in our experimental setup.

Review samples are created such that they follow the variation of aspect and sentiment mentions in the sampled summary. Specifically, we use a Dirichlet distribution, the conjugate prior of the multinomial distribution, to sample N pairs of aspect and sentiment distributions. Given summary y and its distributions $p(a)$ and $p(s)$, the ith pair of aspect and sentiment distributions $\{(p_i(a)p_i(s))\}$, $1 \leq i \leq N$ is sampled as:

$$p_i(a) \sim \text{Dirichlet}(\alpha_a * p(a)) \tag{11}$$

$$p_i(s) \sim \text{Dirichlet}(\alpha_s * p(s)) \tag{12}$$

where α_a and α_s are constants which control the variance of the distributions sampled from the Dirichlet. When α values are small, $p(a)$ and $p(s)$ will look more different from the distribution of the summary, and when α values are larger, the sampled distributions will look more similar to the summary. We provide samples with varying α values in the Appendix. Sampling from the Dirichlet ensures that the average of the sampled distribution equals that of the summary us allowing to control how the synthetic dataset is created modulating how aspect and sentiment are represented.

Finally, for each sampled pair $(p_i(a), p_i(s))$, we run a nearest neighbor search over the corpus to find the review $\mathbf{x}_i$ with the most similar pair of distributions. We use Hellinger (1909) distance to quantify the similarity between two distributions, i.e.. $sim(p, q) = \|\sqrt{p} - \sqrt{q}\|_2 / \sqrt{2}$ (we take the average of the similarity scores between aspect and sentiment distributions). This results to an instance within dataset $\mathbb{D}$, where $\mathbf{X} = \{x_1, ..., x_N\}$ is the set of reviews for summary $\mathbf{y}$. We repeat this process multiple times to obtain a large-scale training dataset.

Opinion Summarization

We use the synthetic dataset $\mathbb{D}$ to train our summarization model which we call PLANSUM and illustrate in Figure 3. A fusion module aggregates token-level encodings in input reviews $\mathbf{X}$ to reduce the number of tokens. The fused encodings are then passed to a decoder that uses the mean aspect and sentiment distributions as a content plan to generate output summary $\mathbf{y}$. We do not employ an encoder in our model, but rather reuse the encodings from the content plan induction model, which improves memory-efficiency in comparison to related architectures (Chu and Liu 2019; Bražinskas, Lapata, and Titov 2019; Amplayo and Lapata 2020). At test time, the same model is used to summarize actual reviews.

Mean and Injective Fusion For each review $\mathbf{x}_i \in \mathbf{X}$ with tokens $\{w_j^{(i)}\}$, we obtain token-level encodings $\{h_j^{(i)}\}$ and probability distributions $p^{(i)}(a)$ and $p^{(i)}(s)$, using Equation (1). We then aggregate these encodings and distributions to collectively represent the set of input reviews.

It is trivial to aggregate aspect and sentiment distributions since the synthetic dataset is by construction such that their average equals to the summary. We thus take their mean as follows:

$$p(a) = \sum\nolimits_i p^{(i)}(a)/N \tag{13}$$

$$p(s) = \sum\nolimits_i p^{(i)}(s)/N \tag{14}$$

It is critical to fuse token embeddings as the number of input tokens can be prohibitively large causing out-of-memory issues. We could fuse token embeddings by aggregating over the same word, especially since multiple reviews are highly redundant. However, simple aggregation methods such as mean and max pooling may be all too effective at eliminating redundancy since they cannot retain information regarding token frequency. This would be problematic for our task, redundancy is an important feature of opinion summarization, and repetition can indicate which aspects are considered important. To mitigate this, we borrow a fusion method from graph neural networks (Xu et al. 2019) that uses an injective function, to effectively discriminate representations of the same token but with different levels of redundancy:

$$h_k = \text{MLP}(e_k + \sum\nolimits_{(i,j):w_j^{(i)}=w_k} h_j^{(i)}) \tag{15}$$

where e_k is a learned embedding for word w_k in the vocabulary.

Decoder with Content Planning Our decoder is an LSTM equipped with attention (Bahdanau, Cho, and Bengio 2014) and copy (Vinyals, Fortunato, and Jaitly 2015) mechanisms, where the aggregated token embeddings $\{h_k\}$ are used as keys. Additionally, at each timestep, the decoder makes use of the aggregated probability distributions $p(a)$ and $p(s)$ as a content plan. This guides the model towards generating correct aspect and sentiment information. Specifically, we use embedding matrices $\mathbf{A}$ and $\mathbf{S}$ from the content plan induction model to obtain aspect and sentiment encodings d_a and d_s, using Equations (5) and (6). We then combine these encodings with the output token y_t at timestep t:

$$y'_t = f(d_a, d_s, y_t) \tag{16}$$

$$s_t = \text{LSTM}(y'_t, s_t) \tag{17}$$

$$p(y_{t+1}) = \text{ATTENDCOPY}(y'_t, s_t, \{h_k\}) \tag{18}$$

where $f(\cdot)$ is a linear function.

Training and Inference We use a maximum likelihood loss to optimize the probability distribution based on summary $\mathbf{y} = \{y_t\}$. We also use an LM-based label smoothing method, which instead of the uniform distribution (Szegedy et al. 2016) uses predictions from BERT (Devlin et al. 2019) as a prior distribution:

$$\hat{y}_t = (1 - \delta) * y_t + \delta * \text{BERT}(y_{-t}) \tag{19}$$

$$\mathcal{L}_{gen} = -\sum\nolimits_t \hat{y}_t \log p(y_t) \tag{20}$$

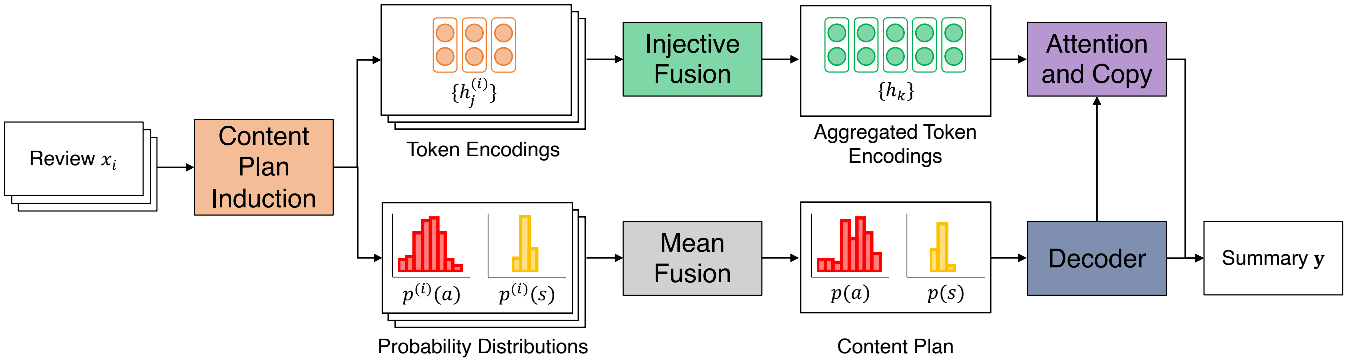

Figure 3: Model architecture of PLANSUM. The content plan is constructed as the average of the aspect and sentiment probability distributions induced by the content plan induction model. It is then passed to the decoder, along with the aggregated token encodings to generate the summary.

Experimental Setup

Datasets

We performed experiments on three opinion summarization benchmarks. These include the Rotten Tomatoes dataset[1] (RT; Wang and Ling 2016) which contains a large set of reviews for various movies written by critics. Each set of reviews has a gold-standard opinion summary written by an editor. However, we do not use ground truth summaries for training, to simulate our unsupervised setting. Our second dataset is Yelp[2] (Chu and Liu 2019) which includes a large training corpus of reviews for businesses without gold-standard summaries, as well as development and test sets where summaries were generated by Amazon Mechanical Turk (AMT) crowdworkers. Finally, the Amazon dataset[3] (Bražinskas, Lapata, and Titov 2019) contains product reviews for four Amazon categories: *Electronics*, *Clothing, Shoes and Jewelry*, *Home and Kitchen*, and *Health and Personal Care*. The development and test partitions come with three gold-standard reference summaries produced by AMT annotators. All datasets include review ratings which we used as sentiment labels: Rotten Tomatoes has binary labels, while Yelp and Amazon have a 1–5 scale.

To create synthetic training data, we sampled candidate summaries using the following constraints: (1) there must be no non-alphanumeric symbols aside from punctuation, (2) there must be no first-person singular pronouns (not used in Yelp/Amazon), and (3) the number of tokens must be between 50–90 (20–50 for RT). We also made sure that sampled reviews and candidate summary discuss the same entity. After applying these constraints we obtained 100k (Yelp), 25k (RT), and 90k (Amazon) review-summary pairs. Statistics of these datasets are reported in Table 1. As can be seen, RT contains the largest number of input reviews but the shortest summaries (22–35 tokens). While Amazon and Yelp have a smaller number of input reviews but longer summaries (66–70.9 and 62.5–59.8 tokens, respectively).

Yelp	Train*	Dev	Test
#summary	100k	100	100
#reviews	8.0	8.0	8.0
#tokens/summary	66.0	70.9	67.3
#tokens/review	65.7	70.3	67.8
corpus size			2,320,800
Rotten Tomatoes	**Train***	**Dev**	**Test**
#summary	25k	536	737
#reviews	72.3	98.0	100.3
#tokens/summary	25.8	23.6	23.8
#tokens/review	22.9	23.5	23.6
corpus size			245,848
Amazon	**Train***	**Dev**	**Test**
#summary	90k	28×3	32×3
#reviews	8.0	8.0	8.0
#tokens/summary	59.8	60.5	62.5
#tokens/review	55.8	56.0	56.0
corpus size			1,175,191

Table 1: Dataset statistics; Train* column refers to the synthetic data we created. Amazon contains three reference summaries (× 3) per instance.

Training Configuration

Across models, we set all hidden dimensions to 256, the dropout rate to 0.1, and batch size to 16. We used the subword tokenizer of BERT (Devlin et al. 2019), which has a 30k token vocabulary trained using WordPiece (Wu et al. 2016). For RT, we follow Wang and Ling (2016) and add a generic label for movie titles during training which we replace with the original title during inference. We used the Adam optimizer (Kingma and Ba 2015) with a learning rate of $3e-4$, l_2 constraint of 3, and warmup of 8,000 steps. We also used dropout (Srivastava et al. 2014) after every non-linear function. For each dataset, we additionally tuned the number of aspects, regularization parameter λ, Dirichlet parameters α_a and α_s, label smoothing parameter δ, and beam search size on the development set. We performed early stopping based on the token-level accuracy of the model, again on the development set. Our model was trained on a single GeForce GTX 1080Ti GPU and is implemented using

[1] http://www.ccs.neu.edu/home/luwang/data.html

[2] https://github.com/sosuperic/MeanSum

[3] https://github.com/ixlan/Copycat-abstractive-Amazon-product-summaries

	Yelp			RT			Amazon		
Model	R1	R2	RL	R1	R2	RL	R1	R2	RL
LexRank	25.50	2.64	13.37	14.88	1.94	10.50	28.74	5.47	16.75
W2vCent	24.61	2.85	13.81	13.93	2.10	10.81	28.73	4.97	17.45
SnCent	25.05	3.09	14.56	15.90	2.01	11.74	30.45	5.40	17.73
BertCent	26.67	3.19	14.67	17.65	2.78	12.78	30.67	5.21	17.76
Opinosis	25.15	2.61	13.54	14.98	3.07	12.19	28.42	4.57	15.50
MeanSum	28.86	3.66	15.91	15.79	1.94	12.26	29.20	4.70	18.15
DenoiseSum	30.14	4.99	17.65	21.26	4.61	16.27	—	—	—
CopyCat	29.47	5.26	18.09	—	—	—	31.97	5.81	**20.16**
PlanSum	**34.79***	**7.01***	**19.74***	**21.77***	**6.18**	**16.98***	**32.87***	**6.12***	19.05

Table 2: Automatic evaluation on Yelp, RT, and Amazon datasets. Extractive/Abstractive models shown in first/second block. Best systems shown in bold and 2nd best systems are underlined; asterisk (*) means there is a significant difference between best and 2nd best systems (based on paired bootstrap resampling; $p < 0.05$).

PyTorch.[4] A more detailed model configuration is described in the Appendix.

Comparison Systems

We compared PlanSum to several previously proposed approaches. Extractive systems include LexRank (Erkan and Radev 2004), a PageRank-like algorithm that selects the most salient sentences from the input, and several variants of a centroid-based (Radev et al. 2004) baseline which selects as summary the review closest to the centroid of a group. Specifically, we present results with different input representations, such as in-domain word2vec (Mikolov et al. 2013) embeddings (W2vCent; Rossiello, Basile, and Semeraro 2017), encodings from Sentiment Neuron (Radford, Józefowicz, and Sutskever 2017), an LSTM-based language model trained on a large review corpus (SnCent; Amplayo and Lapata 2020), and encodings from BERT (Devlin et al. 2019), a large transformer-based language model trained using huge amounts of data (BertCent).

Abstractive comparison systems include Opinosis (Ganesan, Zhai, and Han 2010), a graph-based method that uses token-level redundancy to generate summaries, MeanSum (Chu and Liu 2019), an autoencoder that generates summaries by reconstructing the mean of review encodings, DenoiseSum (Amplayo and Lapata 2020), a denoising model that treats non-salient information as noise and removes it to generate a summary, and CopyCat (Bražinskas, Lapata, and Titov 2019), a hierarchical variational autoencoder which learns a latent code of the summary.

Model	Yelp	RT	Amazon
PlanSum	19.74	16.98	19.05
No disentangling	18.83	16.09	18.52
No regularization	19.00	16.85	18.92
Random sampling	19.22	16.61	18.70
Similarity sampling	19.38	15.06	18.31
No content plan	19.03	16.56	18.28
Mean token fusion	18.72	16.76	18.57
Uniform label prior	18.80	16.77	18.94

Table 3: PlanSum with less expressive plan induction (second block), using alternative review sampling methods (third block), and without some modules (fourth block). See Appendix for more detailed comparisons.

Results

Automatic Evaluation We evaluated the quality of opinion summaries using F_1 ROUGE (Lin and Hovy 2003). Unigram and bigram overlap (ROUGE-1 and ROUGE-2) are a proxy for assessing informativeness while the longest common subsequence (ROUGE-L) measures fluency.

Our results are summarized in Table 2. Among extractive models, BertCent performs best, indicating that representations from large transformer-based language models can be used as a simple method to produce good extractive summaries. Extractive models, however, are consistently worse than neural-based abstractive models. Amongst the latter, PlanSum performs best across datasets and metrics save in terms of ROUGE-L on Amazon. The slight better performance of CopyCat suggests that the use of a VAE objective may also be beneficial for our model, however we leave this to future work. Especially on Yelp, we observe a large improvement, with an increase of 5.32, 1.75, and 1.65 points in ROUGE-1/2/L over the best comparison systems. Our unsupervised model is comparable to the best supervised model (Amplayo and Lapata 2019), performing 0.58 points better on ROUGE-1 and 0.82 points worse on ROUGE-L. We show examples of system output for our model and comparison systems in the Appendix.

We present in Table 3 various ablation studies on the three datasets, which assess the contribution of different model components. Our experiments confirm that aspect and sentiment disentanglement and embedding regularization in the content plan induction module improve performance. Moreover, our dataset creation method is better than random or similarity sampling. This is especially the case on Rotten Tomatoes, where there is an 1.92 decrease in ROUGE-L. Rotten Tomatoes differs from Amazon and Yelp in that the input reviews are multiple (in the excess of 50) and thus contains more variety which our content planning approach manages to capture and reproduce in generating the synthetic data. Finally, we show that the use of the content plan,

[4] Our code can be downloaded from https://github.com/rktamplayo/PlanSum.

PlanSum
This is a great place to hang out with friends. The staff is very friendly and helpful. They have a lot of different beers to choose from and the beer selection is great. I'm not a big fan of beers but this place has some good selections. If you're in the mood for a beer and a fun atmosphere, this will be the place for you.
Random Sampling
This is a great place to hang out with friends and family. The beer selection is great, and the atmosphere is very nice. I've been here a few times and have never had a bad experience. It's a fun place for a group of friends or groups.
Similarity Sampling
This is *a great place to go if you're in the area.* It's a cool place for a night out, but it is well worth it. The atmosphere is great and the staff is always friendly. I'm not sure if I will go back.
No Plan
This is a great place to hang out with friends. The staff is very friendly and the beer selection is great. I've had a couple of beers and they have a good selection of beer and beer. *It's a little pricey but it is worth the wait.*

Table 4: Yelp summaries generated by PlanSum and variants thereof. Aspects also mentioned in the gold summary (not shown to save space) are in color (atmosphere, staff, and beer), all other aspects are *italicized*.

Yelp	Asp	Sen	Coh	Gam
BertCent	−9.0	−1.5	−2.9	−7.4
DenoiseSum	−11.3	−11.1	−6.5	−10.6
CopyCat	−5.8	−15.0	−15.8	−10.0
PlanSum	**3.9**	**6.9**	**5.7**	**7.0**
Gold	22.2	20.7	19.4	20.9
Rotten Tomatoes	Asp	Sen	Coh	Gam
BertCent	−8.4	−12.2	−6.9	−4.0$^{\not *}$
DenoiseSum	−31.1	−6.9$^{\not *}$	−25.1	−17.3
CopyCat	—	—	—	−10.0
PlanSum	**10.7**	**1.3**	**2.2**	**−2.2**
Gold	28.9	20.4	29.8	23.6
Amazon	Asp	Sen	Coh	Gam
BertCent	−10.7	**−3.1**$^{\not *}$	−7.1	−9.1$^{\not *}$
DenoiseSum	—	—	—	—
CopyCat	−9.8	−18.9	−10.2	−12.22
PlanSum	**0.0**	−6.4	**7.1**	**−1.8**
Gold	20.4	28.4	10.2	23.1

Table 5: Best-worst scaling: aspect- and sentiment-based informativeness (Asp and Sen), coherence (Coh), grammaticality (Gram). All pairwise differences between PlanSum and other systems are significant, except when there is an asterisk ($\not *$), using a one-way ANOVA with posthoc Tukey HSD tests ($p < 0.05$).

injective fusion module, and the LM-based label smoothing all increase generation performance.

In Table 4 we show how content planning modulates summary output. We present a summary produced by PlanSum and variants without a content plan during synthetic data creation (see Random and Similarity Sampling) and in the summarization model (No Plan). Summaries without any planning whatsoever either miss out on salient aspects, or focus on aspects that do not reach consensus (i.e., aspect mentions absent from the summary).

Human Evaluation We also conducted a judgment elicitation study using the Amazon Mechanical Turk crowdsourcing platform. We assessed the quality of system summaries using Best-Worst Scaling (Louviere, Flynn, and Marley 2015). Specifically, we asked participants to select the *best* and *worst* among system summaries taking into account how much they deviated from given input reviews in terms of four criteria. The first two criteria assess informativeness and ask crowdworkers to select a summary based on whether it mentions the majority of *aspects* discussed in the original reviews and agrees with their overall *sentiment*. We also evaluate summaries in terms of *coherence* (i.e., is the summary easy to read and does it follow a natural ordering of facts?), and *grammaticality* (i.e., is the summary fluent?). We randomly selected 30 instances from the test set. For Rotten Tomatoes, we filtered out instances where the number of input reviews exceeded 30 so that participants could read the reviews in a timely fashion. We collected three judgments for each comparison. The order of summaries was randomized per participant. A rating per system was computed as the percentage of times it was chosen as best minus the percentage of times it was selected as worst.

We compared summaries produced by the BertCent extractive baseline, our model PlanSum, and two competitive unsupervised abstractive systems, DenoiseSum (Amplayo and Lapata 2020) and CopyCat (Bražinskas, Lapata, and Titov 2019). We also included human-authored summaries as an upper bound. The ratings are reported in Table 5. Overall, the gold summaries were consistently rated the highest on all criteria. Among the system summaries, PlanSum was rated the best in terms of all criteria, except on sentiment-based informativeness for Amazon, where BertCent was given the highest rating. BertCent surprisingly was rated higher than the other abstractive systems. We inspected the summaries produced by these systems and found that CopyCat summaries are more positive-oriented and DenoiseSum summaries contain more grammatical errors, as also reflected in the ratings. We posit that these errors are possibly due to the use of random sampling and noising functions, respectively, when creating the synthetic dataset. We show examples of generated summaries in the Appendix.

Conclusions

In this work we considered the use of aspect and sentiment distributions as a content plan for unsupervised opinion summarization which we argued leads to higher quality summaries and allows for the creation of naturalistic synthetic datasets. Extensive automatic and human-based evaluation showed that our model outperforms competitive systems on three benchmarks with varying characteristics. In the future, we plan to explore personalization in opinion summarization, where the content plan can be used to control generation towards more aspect- or sentiment-specific information. We also plan to apply the techniques in this paper to domains where documents are longer (e.g., news articles).

Acknowledgments

We thank the anonymous reviewers for their feedback. We gratefully acknowledge the support of support of the European Research Council (Lapata, award number 681760) The first author is supported by a Google PhD Fellowship.

References

Amplayo, R. K.; and Lapata, M. 2019. Informative and controllable opinion summarization. *arXiv preprint arXiv:1909.02322* .

Amplayo, R. K.; and Lapata, M. 2020. Unsupervised Opinion Summarization with Noising and Denoising. *arXiv preprint arXiv:2004.10150* .

Amplayo, R. K.; Lim, S.; and Hwang, S.-w. 2018. Entity Commonsense Representation for Neural Abstractive Summarization. In *NAACL*, 697–707. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-1064.

Angelidis, S.; and Lapata, M. 2018. Summarizing Opinions: Aspect Extraction Meets Sentiment Prediction and They Are Both Weakly Supervised. In *EMNLP*, 3675–3686. Brussels, Belgium: Association for Computational Linguistics. doi: 10.18653/v1/D18-1403.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*. San Diego, California.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A.; Jozefowicz, R.; and Bengio, S. 2016. Generating Sentences from a Continuous Space. In *CoNLL*, 10–21. Berlin, Germany: Association for Computational Linguistics. doi:10.18653/v1/K16-1002.

Bražinskas, A.; Lapata, M.; and Titov, I. 2019. Unsupervised Multi-Document Opinion Summarization as Copycat-Review Generation. *arXiv preprint arXiv:1911.02247* .

Carenini, G.; Cheung, J. C. K.; and Pauls, A. 2013. Mutli-document Summarization of Evaluative Text. *Computational Intelligence* 29(4): 545–576.

Carenini, G.; Ng, R.; and Pauls, A. 2006. Multi-Document Summarization of Evaluative Text. In *EACL*. Trento, Italy: Association for Computational Linguistics.

Chu, E.; and Liu, P. 2019. MeanSum: A Neural Model for Unsupervised Multi-Document Abstractive Summarization. In Chaudhuri, K.; and Salakhutdinov, R., eds., *ICML*, volume 97 of *Proceedings of Machine Learning Research*, 1223–1232. Long Beach, California, USA: PMLR.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/N19-1423.

Di Fabbrizio, G.; Stent, A.; and Gaizauskas, R. 2014. A Hybrid Approach to Multi-document Summarization of Opinions in Reviews. In *INLG*, 54–63. Philadelphia, Pennsylvania.

Erkan, G.; and Radev, D. R. 2004. LexRank: Graph-based Lexical Centrality As Salience in Text Summarization. *Journal of Artificial Intelligence Research* 22(1): 457–479. ISSN 1076-9757.

Fevry, T.; and Phang, J. 2018. Unsupervised Sentence Compression using Denoising Auto-Encoders. In *CoNLL*, 413–422. Brussels, Belgium: Association for Computational Linguistics. doi:10.18653/v1/K18-1040.

Freitag, M.; and Roy, S. 2018. Unsupervised Natural Language Generation with Denoising Autoencoders. In *EMNLP*, 3922–3929. Brussels, Belgium: Association for Computational Linguistics. doi:10.18653/v1/D18-1426.

Ganesan, K.; Zhai, C.; and Han, J. 2010. Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions. In *COLING*, 340–348. Beijing, China: Coling 2010 Organizing Committee.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; and Lempitsky, V. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research* 17(59): 1–35.

Gehrmann, S.; Dai, F.; Elder, H.; and Rush, A. 2018. End-to-End Content and Plan Selection for Data-to-Text Generation. In *INLG*, 46–56. Tilburg University, The Netherlands: Association for Computational Linguistics. doi:10.18653/v1/W18-6505.

He, R.; Lee, W. S.; Ng, H. T.; and Dahlmeier, D. 2017. An Unsupervised Neural Attention Model for Aspect Extraction. In *ACL*, 388–397. Vancouver, Canada: Association for Computational Linguistics. doi:10.18653/v1/P17-1036.

Hellinger, E. 1909. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1909(136): 210–271.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Comput.* 9(8): 1735–1780. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735.

Hu, M.; and Liu, B. 2006. Opinion Extraction and Summarization on the Web. In *AAAI*, AAAI'06, 1621–1624. AAAI Press. ISBN 978-1-57735-281-5.

Hua, X.; and Wang, L. 2019. Sentence-Level Content Planning and Style Specification for Neural Text Generation. In *EMNLP-IJCNLP*, 591–602. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1055.

Kan, M.-Y.; and McKeown, K. R. 2002. Corpus-trained Text Generation for Summarization. In *INLG*, 1–8. Harriman, New York, USA: Association for Computational Linguistics.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*. San Diego, California.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *ICLR*. Banff, AB.

Ku, L.-W.; Liang, Y.-T.; and Chen, H.-H. 2006. Opinion Extraction, Summarization and Tracking in News and Blog Corpora. In *AAAI-CAAW*, 100–107.

Kukich, K. 1983. Design of a Knowledge-Based Report Generator. In *ACL*, 145–150. Cambridge, Massachusetts, USA: Association for Computational Linguistics. doi:10.3115/981311.981340.

Lin, C.-Y.; and Hovy, E. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *NAACL*, 150–157.

Louviere, J. J.; Flynn, T. N.; and Marley, A. A. J. 2015. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press. doi:10.1017/CBO9781107337855.

McKeown, K. 1985. *Text Generation*. Studies in Natural Language Processing. Cambridge University Press. doi:10.1017/CBO9780511620751.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *NIPS*, NIPS'13, 3111–3119. Red Hook, NY, USA: Curran Associates Inc.

Moryossef, A.; Goldberg, Y.; and Dagan, I. 2019. Step-by-Step: Separating Planning from Realization in Neural Data-to-Text Generation. In *NAACL*, 2267–2277. Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/N19-1236.

Mukherjee, A.; and Liu, B. 2012. Aspect Extraction through Semi-Supervised Modeling. In *ACL*, 339–348. Jeju Island, Korea: Association for Computational Linguistics.

Pang, B.; and Lee, L. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* 2(1-2): 1–135. ISSN 1554-0669. doi:10.1561/1500000011.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In *EMNLP*, EMNLP '02, 79–86. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1118693.1118704.

Paul, M.; Zhai, C.; and Girju, R. 2010. Summarizing Contrastive Viewpoints in Opinionated Text. In *EMNLP*, 66–76. Cambridge, MA: Association for Computational Linguistics.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; Hoste, V.; Apidianaki, M.; Tannier, X.; Loukachevitch, N.; Kotelnikov, E.; Bel, N.; Jiménez-Zafra, S. M.; and Eryiğit, G. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *(SemEval)*, 19–30. San Diego, California: Association for Computational Linguistics. doi:10.18653/v1/S16-1002.

Puduppully, R.; Dong, L.; and Lapata, M. 2019. Data-to-Text Generation with Content Selection and Planning. In *AAAI*, 6908–6915. doi:10.1609/aaai.v33i01.33016908.

Radev, D. R.; Jing, H.; Styś, M.; and Tam, D. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management* 40(6): 919–938.

Radford, A.; Józefowicz, R.; and Sutskever, I. 2017. Learning to Generate Reviews and Discovering Sentiment. *CoRR* abs/1704.01444.

Rossiello, G.; Basile, P.; and Semeraro, G. 2017. Centroid-based Text Summarization through Compositionality of Word Embeddings. In *MultiLing*, 12–21. Valencia, Spain: Association for Computational Linguistics. doi:10.18653/v1/W17-1003.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *EMNLP*, 379–389. Lisbon, Portugal: Association for Computational Linguistics. doi:10.18653/v1/D15-1044.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *ACL*, 1073–1083. Vancouver, Canada: Association for Computational Linguistics. doi:10.18653/v1/P17-1099.

Sharma, E.; Huang, L.; Hu, Z.; and Wang, L. 2019. An Entity-Driven Framework for Abstractive Summarization. In *EMNLP-IJCNLP*, 3280–3291. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1323.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 15(1): 1929–1958. ISSN 1532-4435.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *NIPS*, 3104–3112. Curran Associates, Inc.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR*.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer Networks. In *NIPS*, NIPS'15, 2692–2700. Cambridge, MA, USA: MIT Press.

Wang, L.; and Ling, W. 2016. Neural Network-Based Abstract Generation for Opinions and Arguments. In *NAACL*, 47–57. San Diego, California: Association for Computational Linguistics. doi:10.18653/v1/N16-1007.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *ICLR*.