Informative and Controllable Opinion Summarization

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

Abstract

Opinion summarization is the task of automatically generating summaries for a set of reviews about a specific target (e.g., a movie or a product). Since the number of reviews for each target can be prohibitively large, neural network-based methods follow a two-stage approach where an *extractive* step first pre-selects a subset of salient opinions and an *abstractive* step creates the summary while conditioning on the extracted subset. However, the extractive model leads to loss of information which may be useful depending on user needs. In this paper we propose a summarization framework that eliminates the need to rely only on pre-selected content and waste possibly useful information, especially when customizing summaries. The framework enables the use of all input reviews by first *condensing* them into multiple dense vectors which serve as input to an abstractive model. We showcase an effective instantiation of our framework which produces more informative summaries and also allows to take user preferences into account using our zero-shot customization technique. Experimental results demonstrate that our model improves the state of the art on the Rotten Tomatoes dataset and generates customized summaries effectively.

"Coach Carter" Reviews
• Samuel L. Jackson plays the real-life coach of a high school basketball team in this solid sports drama ...
• Great performance by Samuel Jackson but predictable as a slam dunk ...
• ... excellent basketball choreography, Coach Carter is fun, hopeful, occasionally silly and, what can I say, inspiring.
Consensus Summary
Even though it's based on a true story, Coach Carter is pretty formulaic stuff, but it's effective and energetic, thanks to a strong central performance from Samuel L. Jackson.
EXTRACT-ABSTRACT Framework
Coach Carter is a preposterously plotted thriller that borrows heavily from other superior films. *(factually incorrect)*
CONDENSE-ABSTRACT Framework
General: An inspirational flick with a healthy dose of message, but it's too predictable.
Customized (acting): An inspirational flick with a healthy dose of humor, Coach Carter is a perceptive sports drama with a standout performance from Samuel L. Jackson.
Customized (plot): A feel-good tale with a healthy dose of heart, Coach Carter is a worthy addition to the basketball system that it's difficult to resist.

Figure 1: Three out of 150 reviews for the movie "Coach Carter", and summaries written by the editor, and generated by a model following the EXTRACT-ABSTRACT approach and the proposed CONDENSE-ABSTRACT framework. The latter produces more informative and factual summaries whilst allowing to control aspects of the generated summary (such as the acting or plot of the movie).

1 Introduction

The proliferation of opinions expressed in online reviews, blogs, and social media has created a pressing need for automated systems which enable customers and companies to make informed decisions without having to absorb large amounts of opinionated text. Opinion summarization is the task of automatically generating summaries for a set of opinions about a specific target (Conrad et al., 2009). Figure 1 shows various reviews about the movie "Coach Carter" and example summaries generated by humans and automatic systems.

The vast majority of previous work (Hu and Liu, 2004) views opinion summarization as the final stage of a three-step process involving: (1) aspect extraction (i.e., finding features pertaining to the target of interest, such as battery life or sound quality); (2) sentiment prediction (i.e., determining the sentiment of the extracted aspects); and (3) summary generation (i.e., presenting the identified opinions to the user). Textual summaries are created following mostly extractive methods which select representative segments (usually sentences) from the source text (Popescu and Etzioni,

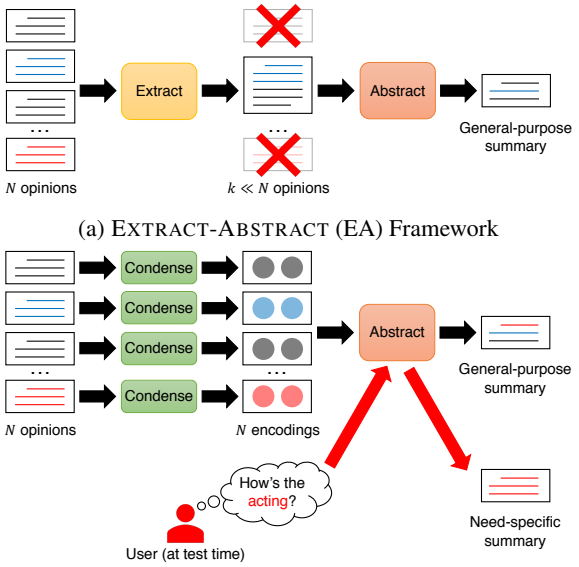
Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2662–2672
April 19 - 23, 2021.

2005; Blair-Goldensohn et al., 2008; Lerman et al., 2009). Despite being less popular, abstractive approaches seem more appropriate for the task at hand as they attempt to generate summaries which are maximally informative and minimally redundant without simply rearranging passages from the original opinions (Ganesan et al., 2010; Carenini et al., 2013; Gerani et al., 2014).

General-purpose summarization approaches have recently shown promising results with end-to-end models which are data-driven and take advantage of the success of sequence-to-sequence neural network architectures. Most approaches (Rush et al., 2015; See et al., 2017) encode documents and then decode the learned representations into an abstractive summary, often by attending to the source input (Bahdanau et al., 2014) and copying words from it (Vinyals et al., 2015). Under this modeling paradigm, it is no longer necessary to identify aspects and their sentiment for the opinion summarization task, as these are learned *indirectly* from training data (i.e., sets of opinions and their corresponding summaries). These models are usually tested on domains where the input is either one document or a small set of documents.

However, the number of input reviews for each target entity tends to be very large (150 for the example in Figure 1). It is therefore practically unfeasible to train a model in an end-to-end fashion, given the memory limitations of modern hardware. As a result, current approaches (Wang and Ling, 2016; Liu et al., 2018; Liu and Lapata, 2019) sacrifice end-to-end elegance in favor of a two-stage framework which we call EXTRACT-ABSTRACT (EA): an *extractive* model first selects a subset of opinions and an *abstractive* model then generates the summary while conditioning on the extracted subset (see Figure 2a). The extractive pass unfortunately has two drawbacks. Firstly, on account of having access to only a small subset of reviews, the summaries can be less informative and inaccurate, as shown in Figure 1. And secondly, user preferences cannot be easily taken into account (e.g., a user may wish to obtain a summary focusing on the acting or plot of a movie as opposed to a general-purpose summary) since more specialized information might have been removed.

In this paper, we propose CONDENSE-ABSTRACT (CA), an alternative two-stage framework which enables the use of *all* input reviews when generating the summary (see



(a) EXTRACT-ABSTRACT (EA) Framework

(b) CONDENSE-ABSTRACT (CA) Framework

Figure 2: Illustration of EA and CA frameworks for opinion summarization. In the CA framework, users can obtain need-specific summaries at test time (e.g., give me a summary focusing on acting).

Figure 2b). The CONDENSE model first represents the input reviews as encodings, aiming to condense their meaning and distill information relating to sentiment and various aspects of the target being reviewed. The ABSTRACT model then fuses these condensed representations into one aggregate encoding and generates an opinion summary from it. We implement a simple yet effective instantiation of the CA framework, using a vanilla autoencoder as the CONDENSE model, and a decoder with attention and copy mechanisms as the ABSTRACT model. We also introduce a zero-shot customization technique allowing users to control important aspects of the generated summary at test time. Our approach enables controllable generation while leveraging the full spectrum of opinions available for a specific target.

We perform experiments on a dataset consisting of movie reviews and opinion summaries elicited from the Rotten Tomatoes website (Wang and Ling, 2016; see Figure 1). Our proposed approach outperforms state-of-the-art models by a large margin using automatic metrics and in a judgment elicitation study. We also verify that our zero-shot customization technique can effectively generate need-specific summaries.

2 Related Work

Most opinion summarization models follow extractive methods (see Kim et al., 2011 and Angelidis and Lapata, 2018 for overviews), with the exception of a few systems which are able to generate novel words and phrases not featured in the source text. Ganesan et al. (2010) propose a graph-based framework for generating concise opinion summaries, while Gerani et al. (2014) represent reviews as discourse trees which they aggregate to a global graph to generate a summary. Other work (Carenini et al., 2013; Mukherjee and Joshi, 2013) takes the distribution of opinions and their aspects into account so as to generate more readable summaries. Di Fabbrizio et al. (2014) present a hybrid system which uses extractive techniques to select salient quotes from the input reviews and embeds them into an abstractive summary to provide evidence for positive or negative opinions.

More recent work has seen the effective application of sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2014) to various abstractive summarization tasks including headline generation (Rush et al., 2015), single- (See et al., 2017; Nallapati et al., 2016), and multi-document summarization (Wang and Ling, 2016; Liu et al., 2018; Liu and Lapata, 2019). Closest to our approach is the work of Wang and Ling (2016) who generate opinion summaries following a two-stage process which first selects/extracts reviews bearing pertinent information, and then generates the summary by conditioning on these reviews. More recent models (Chu and Liu, 2019; Bražinskas et al., 2020; Amplayo and Lapata, 2020) perform opinion summarization in an unsupervised way. However, these are mostly done on toy datasets (Chu and Liu, 2019), typically with a small number of reviews per target entity.

Our proposed framework works better on real-world datasets with a large number of reviews, since it eliminates the need to rely only on pre-selected salient reviews which we argue leads to information loss and subsequently less customizable generation. Instead, our model first *condenses* the source reviews into multiple dense vectors which serve as input to a decoder to generate an abstractive summary. Beyond producing more informative summaries, we demonstrate that our approach also allows to customize them. Recent conditional generation models have focused on controlling various aspects of the output such as politeness (Sennrich et al., 2016), length (Kikuchi et al., 2016), content (Fan et al., 2018), or style (Ficler and Goldberg, 2017). In contrast, our zero-shot customization technique requires neither training examples of documents and corresponding (customized) summaries nor specialized pre-processing to encode which tokens in the input might give rise to customization.

3 CONDENSE-ABSTRACT Framework

We propose an alternative to the EXTRACT-ABSTRACT (EA) approach which enables the use of all input reviews when generating the summary. Figure 2b illustrates our proposed CONDENSE-ABSTRACT (CA) framework. In lieu of an integrated encoder-decoder, we generate summaries using two separate models. The CONDENSE model returns review encodings for N input reviews, while the ABSTRACT model uses these encodings to create an abstractive summary. This two-step approach has two advantages for multi-document summarization. Firstly, CA-based models are more space-efficient, since the set of N reviews is not treated as one large instance but as N separate instances when training the CONDENSE model. And secondly, it is possible to generate maximally informative and customizable summaries targeting specific aspects of the input since the ABSTRACT model operates over the encodings of *all* available reviews.

In the following subsections, we explain how we instantiate a model using the CA framework, which we call CONDASUM, with an LSTM-based[1] vanilla autoencoder (CONDENSE model) and a decoder with attention and copy mechanisms (ABSTRACT model).

3.1 The CONDENSE Model

Let $\mathcal{D}$ denote a cluster of N reviews about a specific target (e.g., a movie or product). For each review $X = \{w_1, w_2, ..., w_M\} \in \mathcal{D}$, the CONDENSE model learns an encoding d, and word-level encodings $h_1, h_2, ..., h_M$. We employ a Bidirectional Long Short Term Memory (BiLSTM) encoder (Hochreiter and Schmidhuber, 1997) as our

[1]We use LSTMs as our text encoder instead of other popular alternatives, such as Transformers (Vaswani et al., 2017), since LSTMs work better on autoencoder architectures, as shown in the literature (Liu et al., 2019; Zhang et al., 2020), as well as during our preliminary experiments.

CONDENSE model:

$$\{\overrightarrow{h}_i, \overleftarrow{h}_i\} = \text{BiLSTM}_f(w_i) \quad (1)$$

$$h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i] \quad d = [\overrightarrow{h}_M; \overleftarrow{h}_1] \quad (2)$$

where $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ are forward and backward hidden states of the BiLSTM at timestep i, and ; denotes concatenation.

Training is performed with a reconstruction objective. We use a separate LSTM as the decoder where the first hidden state z_0 is set to d. Words w'_t are generated using a softmax classifier:

$$z_t = \text{LSTM}_d(w'_{t-1}, z_{t-1}) \quad (3)$$

$$p(w'_t) = \text{softmax}(W z_t + b) \quad (4)$$

The auto-encoder is trained with a maximum likelihood loss:

$$\mathcal{L}_{condense} = -\sum_{t=1}^{M} \log p(w_t) \quad (5)$$

Once training has taken place, we use the CONDENSE model to obtain N pairs of review encodings $\{d_i\}$ and word-level encodings $\{h_{i,1}, h_{i,2}, ..., h_{i,M}\}$, $1 \leq i \leq N$ as representations for the reviews in $\mathcal{D}$.

3.2 The ABSTRACT Model

The ABSTRACT model first fuses the multiple encodings obtained from the CONDENSE stage and then generates a summary using a decoder.

Multi-source Fusion We aggregate N pairs of review encodings $\{d_i\}$ and word-level encodings $\{h_{i,1}, h_{i,2}, ..., h_{i,M}\}$, $1 \leq i \leq N$ into a single pair of review encoding d' and word-level encodings $h'_1, h'_2, ..., h'_V$, where V is the number of total unique tokens in the input.

Review encodings are fused using an attentive pooling method which gives more weight to important reviews. Specifically, we learn a set of weight vectors $a_i \in \mathbb{R}^{D_d}$, where D_d is the dimension of d_i, to weight-sum the review encodings:

$$\bar{d} = \sum_i d_i / N \quad (6)$$

$$a_i = \text{softmax}(d_i^\top W_p \bar{d}) \quad (7)$$

$$d' = \sum_i a_i * d_i \quad (8)$$

where the mean encoding $\bar{d}$ is used as the query vector, and $W_p \in \mathbb{R}^{D_d \times D_d \times D_d}$ is a learned tensor.

We also fuse word-level encodings, since the same words may appear in multiple reviews. To do this, we simply average all encodings of the same word, if multiple tokens of the word exist:

$$h'_j = \sum_{(i,k):w_{i,k}=w_j} h_{i,k} / V_{w_j} \quad (9)$$

where V_{w_j} is the number of tokens for word w_j in the input.

Decoder The decoder generates summaries conditioned on the fused review encoding d' and word-level encodings $h'_1, h'_2, ..., h'_V$. We use a simple LSTM decoder enhanced with attention (Bahdanau et al., 2014) and copy mechanisms (Vinyals et al., 2015). We set the first hidden state s_0 to d', and run an LSTM to calculate the current hidden state using the previous hidden state s_{t-1} and word y'_{t-1} at time step t:

$$s_t = \text{LSTM}(y'_{t-1}, s_{t-1}) \quad (10)$$

At each time step t, we use an attention mechanism over word-level encodings to output the attention weight vector a_t and context vector c_t:

$$e_t^i = v^\top \tanh(W_h h'_i + W_s s_t + b_a) \quad (11)$$

$$a_t = \text{softmax}(e_t) \quad (12)$$

$$c_t = \sum_i a_t^i * h'_t \quad (13)$$

Finally, we employ a copy mechanism over the input words to output the final word probability $p(y'_t)$ as a weighted sum over the generation probability $p_g(y'_t)$ and the copy probability $p_c(y'_t)$:

$$p_g(y'_t) = \text{softmax}(W_g[s_t; c_t] + b_g) \quad (14)$$

$$\sigma_t = \sigma(v_s^\top s_t + v_c^\top c_t + v_y^\top y'_t) \quad (15)$$

$$p_c(y'_t) = \sum_{i:y'_i=y'_t} a_t^i \quad (16)$$

$$p(y'_t) = \sigma_t * p_g(y'_t) + (1 - \sigma_t) * p_c(y'_t) \quad (17)$$

where W, v, and b are learned parameters, and t is the current timestep.

Salience-biased Extracts The model presented so far has no explicit mechanism to encourage salience among reviews. We direct the decoder towards salient reviews by incorporating information from an extractive step. Specifically, we use BERTCENT, a centroid-based (Radev et al., 2000) document extraction method that obtains document representations by resorting to BERT (Devlin et al., 2019).

BERTCENT can be simply described as follows. Firstly, given a review, we obtain its encoding as the

average of its token encodings obtained from BERT. We then take the average of the review encodings and treat it as the *centroid* of the input reviews, which approximately represents the information that is considered salient. We select the top k reviews whose encodings are the nearest neigbors to the centroid. The selected reviews are concatenated into a long sequence and encoded using a separate BiLSTM whose output serves as input to an LSTM decoder. This decoder generates a *salience-biased* hidden state r_t. We then update hidden state s_t in Equation (10) as $s_t = [s_t; r_t]$.

Using these extracts, we still take all input reviews into account, while acknowledging that some might be more descriptive than others. This module is a key component to generating *general-purpose* opinion summaries, where a set of aspects is deemed more salient than others (e.g., in general, people care more about the plot rather than the special effects of a movie). However, this extractive module may hurt the customizability of the model (e.g., generating *need-specific* summaries, details explained in Section 3.3), which we show in our experiments in Section 5.

Training We use two objective functions to train the ABSTRACT model. Firstly, we use a maximum likelihood loss to optimize the generation probability distribution $p(y'_t)$ based on gold summaries $Y = \{y_1, y_2, ..., y_L\}$ provided at training time:

$$\mathcal{L}_{generate} = -\sum_{t=1}^{L} \log p(y_t) \qquad (18)$$

Secondly, we propose a way to introduce supervision and guide the attention pooling weights W_p in Equation (7) when fusing the review encodings. Our motivation is that the resulting fused encoding d' should be roughly equivalent to the encoding of summary y, which can be calculated as $z = \text{CONDENSE}(y)$. Specifically, we use a hinge loss that maximizes the inner product between d' and z and simultaneously minimizes the inner product between d' and n_i, where n_i is the encoding of one of five randomly sampled negative summaries:

$$\mathcal{L}_{fuse} = \sum_{i=1}^{5} \max(0, 1 - d'z + d'n_i) \qquad (19)$$

The final objective is then the sum of both loss functions:

$$\mathcal{L}_{abstract} = \mathcal{L}_{generate} + \mathcal{L}_{fuse} \qquad (20)$$

3.3 Zero-shot Customization

At test time, we can either generate a general-purpose summary or a *need-specific* summary. To generate the former, we run the trained model as is and use beam search to find the sequence of words with the highest cumulative probability. To generate the latter, we employ the following simple technique that revises the query vector $\bar{d}$ in Equation (6).

More concretely, in the movie review domain, users might wish to obtain a summary that focuses on a specific sentiment (positive or negative) or aspect (e.g., acting, plot, etc.) of a movie. In a different domain, users might care about the price of a product, its comfort, and so on. Since these summaries are not available at training time, we undertake such customization without requiring access to need-specific summaries. Instead, at test time, we assume access to background reviews to represent the user need. For example, if we wish to generate a positive summary, our method requires a set of reviews with positive sentiment. This is an easy and practical way to approximately provide the model some background on how sentiment is communicated in a review.

We use these background reviews conveying a user need x (e.g., acting, plot, positive or negative sentiment) in the multi-source fusion module to attend more to input reviews related to x. Let C_x denote the set of background reviews. We obtain a new query vector $\hat{d} = \sum_{c=1}^{|C_x|} d_c/|C_x|$, where d_c is the encoding of the c'th review in C_x, calculated using the CONDENSE model. This simple change allows the model to focus on input reviews with semantics similar to the user's need as conveyed by the background reviews C_x. The new query vector $\hat{d}$ is used instead of $\bar{d}$ to obtain review encoding d' (see Equation (6)).

4 Experimental Setup

Dataset We performed experiments on the Rotten Tomatoes dataset[2] provided in Wang and Ling (2016). It contains 3,731 movies; for each movie we are given a large set of reviews written by professional critics and users and a gold-standard consensus summary written by an editor (see an example in Figure 1). We report the dataset statistics in Table 1. Following previous work (Wang and Ling, 2016), we used a generic label for movie

[2] http://www.ccs.neu.edu/home/luwang/publications.html

	Train	Dev	Test
#movies	2,458	536	737
#reviews/movie	100.0	98.0	100.3
#tokens/review	23.6	23.5	23.6
#tokens/summary	23.8	23.6	23.8

Table 1: Dataset statistics of Rotten Tomatoes.

titles during training which we replace with the original titles during inference.

Training Configuration For all experiments, our model used word embeddings with 128 dimensions, pretrained using GloVe (Pennington et al., 2014). We set the dimensions of all hidden vectors to 256 and the batch size to 8. For decoding summaries, we use a length-normalized beam search with beam size of 5. We applied dropout (Srivastava et al., 2014) at a rate of 0.5. The model was trained using the Adam optimizer (Kingma and Ba, 2015) with default parameters and l_2 constraint (Hinton et al., 2012) of 2. We performed early stopping based on model performance on the development set. Our model is implemented in PyTorch[3].

Comparison Systems We compare our approach against two types of methods: one-pass methods and methods that use the EA framework. One-pass methods include (a) LexRank (Erkan and Radev, 2004), a PageRank-like summarization algorithm which generates a summary by selecting the n most salient units, until the length of the target summary is reached; (b) Opinosis (Ganesan et al., 2010), a graph-based abstractive summarizer that generates concise summaries of highly redundant opinions; (c) SummaRunner (Nallapati et al., 2017), a supervised neural extractive model where each review is classified as to whether it should be part of the summary or not; and (d) BertCent, a centroid-based method discussed in Section 3.2 that selects $k = 1$ review nearest to the centroid.

EA-based methods include (g) Regress+S2S (Wang and Ling, 2016), an instantiation of the EA framework where a ridge regression model with hand-engineered features implements the Extract model, while an attention-based sequence-to-sequence neural network is the Abstract model; (h) BertCent+S2S, our implementation of an EA-based system which uses BertCent instead of Regress as the Extract model; and (i) BertCent+PtGen, the same model as (h) but enhanced with a copy mechanism (Vinyals et al., 2015). For all extractive steps, we set $k = 5$, which is tuned on the development set.

[3]Our code can be downloaded from https://github.com/rktamplayo/CondaSum.

5 Results

Automatic Evaluation We considered two evaluation metrics which are also reported in Wang and Ling (2016): METEOR (Denkowski and Lavie, 2014), a recall-oriented metric that rewards matching stems, synonyms, and paraphrases, and ROUGE-SU4 (Lin, 2004) which is calculated as the recall of unigrams and skip-bigrams up to four words. We also report F_1-scores for ROUGE-1/2/L (Lin, 2004). Unigram and bigram overlap (ROUGE-1 and ROUGE-2) are a proxy for assessing informativenes while the longest common subsequence (ROUGE-L) measures fluency.

Our results are presented in Table 2. Among one-pass systems, the extractive model BertCent performs the best; despite being unsupervised and extractive, it benefits from the ability of large neural language models to learn general-purpose representations. When used in EA-based systems, BertCent also improves the system performance, where BertCent+PtGen performs the best. Interestingly, BertCent performs better than BertCent+PtGen in terms of METEOR and ROUGE-SU4, while the latter performs better in terms of ROUGE-1/2/L. Our CA-based model CondaSum outperforms all other models across all metrics, showing that exploiting information about all reviews helps in improving performance.

We present in Table 3 various ablation studies, which assess the contribution of different model components. Results confirm that our multi-source fusion method and the fusion loss improve performance. Morevoer, using BertCent for the salient-biased extractive step is better than no extractive step or using SummaRunner, which is a weaker extractive model. Both multi-source fusion and salient-biased extracts help create better general-purpose summaries; the former learns which reviews to focus on while the latter explicitly selects the most important ones.

Human Evaluation In addition to automatic evaluation, we also assessed system output by eliciting human judgments. Participants compared summaries produced from the best extractive baseline (BertCent), the best EA system (BertCent+PtGen), and our model Conda-

Model	METEOR	ROUGE-SU4	ROUGE-1	ROUGE-2	ROUGE-L
LEXRANK*	5.59	3.98	14.88	1.94	10.50
OPINOSIS*	6.07	4.90	14.98	3.07	12.19
SUMMARUNNER	7.44	5.50	15.86	2.55	12.15
BERTCENT	8.89	7.13	17.65	2.78	12.78
REGRESS+S2S*	6.51	5.70	—	—	—
BERTCENT+S2S	7.42	6.61	17.59	7.34	15.83
BERTCENT+PTGEN	8.15	6.99	19.71	7.43	17.25
CONDASUM	**8.90**	**7.79**	**22.49**	**7.65**	**18.47**

Table 2: Automatic evaluation results on models trained on the original training data. Models whose METEOR and ROUGE-SU4 results are taken from Wang and Ling (2016) are marked with an asterisk *. Best performing results per metric are **boldfaced**.

Model	ROUGE-L
CONDASUM	18.47
Mean document fusion	16.69
No fusion loss	15.10
No salience-biased extracts	16.44
SUMMARUNNER extracts	17.80

Table 3: ROUGE-L of CONDASUM with less effective document fusion method (second block) and without using our salience-biased extractive step (third block). See Appendix for more detailed comparisons.

Model	Inf	Corr	Gram
BERTCENT+PTGEN	-0.263	-0.358	-0.152*
BERTCENT	-0.179	-0.112	-0.102*
CONDASUM	-0.042	0.021	-0.078
GOLD	0.483	0.448	0.331

Table 4: Best-worst scaling scores on informativeness (Inf), correctness (Corr) and grammaticality (Gram). All pairwise systems differences between CONDASUM and other system summaries are significant, except the values marked with asterisk (*), based on a one-way ANOVA with posthoc Tukey HSD tests ($p < 0.05$).

SUM, respectively. As an upper bound, we also included GOLD standard summaries.

The study was conducted on the Amazon Mechanical Turk platform using Best-Worst Scaling (BWS; Louviere et al., 2015), a less labor-intensive alternative to paired comparisons that has been shown to produce more reliable results than rating scales (Kiritchenko and Mohammad, 2017). Specifically, participants were shown the movie title and basic background information (i.e., synopsis, release year, genre, director, and cast). They were also presented with three system summaries and asked to select the *best* and *worst* among them according to three criteria: *Informativeness* (i.e., does the summary convey opinions about specific aspects of the movie in a concise manner?), *Correctness* (i.e., is the information in the summary factually accurate and corresponding to the information given about the movie?), and *Grammaticality* (i.e., is the summary fluent and grammatical?). Examples of summaries are shown in Figure 1 and more can be found in the Appendix. We randomly selected 50 movies from the test set and compared all possible combinations of summary triples for each movie. We collected three judgments for each comparison. The order of summaries and movies was randomized per participant.

The scores are computed as the percentage of times it was chosen as best minus the percentage of times it was selected as worst. The scores range from -1 (worst) to 1 (best) and are shown in Table 4. Perhaps unsurprisingly, the human-generated gold summaries were considered best, whereas our model CONDASUM was ranked second, indicating that humans find its output more informative, correct, and grammatical compared to other systems. BERTCENT was ranked third followed by BERTCENT+PTGEN. We inspected the summaries produced by the latter system and found they were factually incorrect bearing little correspondence to the movie (examples shown in the Appendix), possibly due to the huge information loss at the extraction stage.

Customizing Summaries We further assessed the ability of CA systems to generate customized summaries at test time. We evaluate CONDASUM models with and without the salience-biased extractive step. The latter model biases summary generation towards the k most salient extracted opinions using an additional extractive module which may

GOLD

Whether you choose to see it as a statement on consumer culture or simply a special effects-heavy popcorn flick, Gremlins is a minor classic.

CONDASUM with extractive step

General: Gremlins is a wholesome, entertaining horror film with an enormous cast of eager stars.
Customized (Positive): Gremlins is a wholesome, entertaining horror film with an enormous cast of eager stars.
Customized (Negative): Gremlins is a wholesome, entertaining horror film with an enormous cast of eager stars.

CONDASUM without extractive step

General: Gremlins may appeal to the dark Christmas horror genre.
Customized (Positive): Gremlins is an intelligent, funny Christmas horror film from Joe Dante's novel.
Customized (Negative): Gremlins is an atrociously-acted project whose unoriginal and ineptly-staged horror film from Joe Dante's novel.

Figure 3: Examples of general-purpose and need-specific opinion summaries for the movie "Gremlins", generated by two versions of CONDASUM. We also show the consensus summary (GOLD). Words/phrases in color highlight aspects pertaining to positive and negative. More examples can be found in the Appendix.

discard information relevant to the user's need. We thus expect this model to be less effective for customization than CONDASUM which makes no assumptions regarding which summaries to consider.

In this experiment, we assume users may wish to control the output summaries in four ways focusing on acting- and plot-related aspects of a movie review, as well as its sentiment, which may be positive or negative. Let CUST(x) be the zero-shot customization technique discussed in the Section 3.3, where x is an information need (i.e., acting, plot, positive, or negative). We sampled a set of background reviews C_x ($|C_x|$=1,000) from a corpus of 1 million reviews covering 7,500 movies from the Rotten Tomatoes website, made available in Ficler and Goldberg (2017). The reviews contain sentiment labels provided by their authors and heuristically classified aspect labels. We then ran CUST(x) using both the CONDASUM models. We show in Figure 3 customized summaries generated by the models.

To determine which system is better at customization, we again conducted a judgment elicitation study on Amazon Mechanical Turk. Participants read a summary which was created by a general-purpose system or its customized variant. They were then asked to decide if the summary is generic or focuses on a specific aspect (plot or acting) and expresses positive, negative, or neutral sentiment. We selected 50 movies (from the test set) which had mixed reviews and collected judgments from three different participants per summary. The summaries were presented in random order per participant.

	with extracts		without extracts	
Customized	No	Yes	No	Yes
Acting	40.3	40.3	42.0	**78.0**
Plot	73.3	75.0	51.3	**76.7**
Positive	66.0	67.7	65.3	**80.0**
Negative	22.7	22.0	20.7	**40.7**

Table 5: Proportion of summaries which mention a specific aspect/sentiment. **Boldfaced** values show a significant increase ($p < 0.01$; using two-sample bootstrap tests) compared to the non-customized system variant. Aspects are not mutually exclusive (e.g. a summary may talk about both acting and plot), thus the total percentage may exceed 100%.

Table 5 shows what participants thought of summaries produced by non-customized systems (see column No) and systems which had customization switched on (see column Yes). Overall, we observe that CONDASUM without the extractive step is able to customize summaries to a great extent. In all cases, crowdworkers perceive a significant increase in the proportion of aspect x when using CUST(x). CONDASUM with the extractive step is unable to generate need-specific summaries, showing no discernible difference between generic and customized summaries. This indicates that the use of an extractive module, which is one of the main components of EA-based approaches, limits the flexibility of the abstractive model to customize summaries based on a user need.

6 Conclusions

We introduced the CONDENSE-ABSTRACT (CA) framework for opinion summarization which eliminates the need to rely only on a small subset of extracted reviews and allows the use of all reviews to generate maximally informative summaries. We presented CONDASUM, an instantiation of this framework and showed in both automatic and human-based evaluation that it is superior to purely extractive models and abstractive models that include an extractive pre-selection stage. We also showed that when an extractive step is not used, our zero-shot customization technique is able to generate need-specific summaries at test time. In the future, we plan to apply the CA framework to other multi-document summarization tasks.

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

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*, pages 1934–1945, Online. 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*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California.

Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George Reis, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. In *Proceedings of the WWW Workshop on NLP Challenges in the Information Explosion Era (NLPIX)*, Beijing, China.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online. Association for Computational Linguistics.

Giuseppe Carenini, Jackie Chi Kit Cheung, and Adam Pauls. 2013. Mutli-document summarization of evaluative text. *Computational Intelligence*, 29(4):545–576.

Eric Chu and Peter Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232, Long Beach, California, USA. PMLR.

Jack G. Conrad, Jochen L. Leidner, Frank Schilder, and Ravi Kondadadi. 2009. Query-based opinion summarization for legal blog entries. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 167–176, New York, NY, USA. ACM.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Giuseppe Di Fabbrizio, Amanda Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 54–63, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348, Beijing, China. Coling 2010 Organizing Committee.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602–1613, Doha, Qatar. Association for Computational Linguistics.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

Hyun Duk Kim, Kavita Ganesan, Parikshit Sondhi, and ChengXiang Zhai. 2011. Comprehensive review of opinion summarization. Technical report, University of Illinois at Urbana-Champaign.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 4th International Conference on Learning Representations*, San Diego, CA.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. 2009. Sentiment summarization: Evaluating and learning user preferences. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 514–522, Athens, Greece. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Peter J Liu, Yu-An Chung, and Jie Ren. 2019. Summae: Zero-shot abstractive text summarization using length-agnostic auto-encoders. *arXiv preprint arXiv:1910.00998*.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by summarizing long sequences. In *Proceedings of the 7th International Conference on Learning Representations*, Vancouver, Canada.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. 2015. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press.

Subhabrata Mukherjee and Sachindra Joshi. 2013. Sentiment aggregation using ConceptNet ontology. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 570–578, Nagoya, Japan. Asian Federation of Natural Language Processing.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 3075–3081. AAAI Press.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 339–346, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, USA. Curran Associates Inc.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2692–2700, Cambridge, MA, USA. MIT Press.

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, San Diego, California. Association for Computational Linguistics.

Xinyuan Zhang, Ruiyi Zhang, Manzil Zaheer, and Amr Ahmed. 2020. Unsupervised abstractive dialogue summarization for tete-a-tetes. *arXiv preprint arXiv:2009.06851*.