Unsupervised Opinion Summarization with Noising and Denoising

Reinald Kim Amplayo and **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

The supervised training of high-capacity models on large datasets containing hundreds of thousands of document-summary pairs is critical to the recent success of deep learning techniques for abstractive summarization. Unfortunately, in most domains (other than news) such training data is not available and cannot be easily sourced. In this paper we enable the use of supervised learning for the setting where there are only documents available (e.g., product or business reviews) without ground truth summaries. We create a synthetic dataset from a corpus of user reviews by sampling a review, pretending it is a summary, and generating noisy versions thereof which we treat as pseudo-review input. We introduce several linguistically motivated noise generation functions and a summarization model which learns to denoise the input and generate the original review. At test time, the model accepts genuine reviews and generates a summary containing salient opinions, treating those that do not reach consensus as noise. Extensive automatic and human evaluation shows that our model brings substantial improvements over both abstractive and extractive baselines.

1 Introduction

The proliferation of massive numbers of online product, service, and merchant reviews has provided strong impetus to develop systems that perform opinion mining automatically (Pang and Lee, 2008). The vast majority of previous work (Hu and Liu, 2006) breaks down the problem of opinion aggregation and summarization into three interrelated tasks involving aspect extraction (Mukherjee and Liu, 2012), sentiment identification (Pang et al., 2002; Pang and Lee, 2004), and summary creation based on *extractive* (Radev et al., 2000; Lu et al., 2009) or *abstractive* methods (Ganesan et al., 2010; Carenini et al., 2013; Gerani et al., 2014; Di Fabbrizio et al., 2014). Although potentially more challenging, abstractive approaches seem more appropriate for generating informative and concise summaries, e.g., by performing various rewrite operations (e.g., deletion of words or phrases and insertion of new ones) which go beyond simply copying and rearranging passages from the original opinions.

Abstractive summarization has enjoyed renewed interest in recent years thanks to the availability of large-scale datasets (Sandhaus, 2008; Hermann et al., 2015; Grusky et al., 2018; Liu et al., 2018; Fabbri et al., 2019) which have driven the development of neural architectures for summarizing single and multiple documents. Several approaches (See et al., 2017; Celikyilmaz et al., 2018; Paulus et al., 2018; Gehrmann et al., 2018; Liu et al., 2018; Perez-Beltrachini et al., 2019; Liu and Lapata, 2019; Wang and Ling, 2016) have shown promising results with sequence-to-sequence models that encode one or several source documents and then decode the learned representations into an abstractive summary.

The supervised training of high-capacity models on large datasets containing hundreds of thousands of document-summary pairs is critical to the recent success of deep learning techniques for abstractive summarization. Unfortunately, in most domains (other than news) such training data is not available and cannot be easily sourced. For instance, manually writing opinion summaries is practically impossible since an annotator must read all available reviews for a given product or service which can be prohibitively many. Moreover, different types of products impose different restrictions on the summaries which might vary in terms of length, or the types of aspects being mentioned, rendering the application of transfer learning techniques (Pan and Yang, 2010) problematic.

Motivated by these issues, Chu and Liu (2019) consider an *unsupervised* learning setting where

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1934–1945
July 5 - 10, 2020.

there are only documents (product or business reviews) available without corresponding summaries. They propose an end-to-end neural model to perform abstractive summarization based on (a) an autoencoder that learns representations for each review and (b) a summarization module which takes the aggregate encoding of reviews as input and learns to generate a summary which is semantically similar to the source documents. Due to the absence of ground truth summaries, the model is not trained to reconstruct the aggregate encoding of reviews, but rather it only learns to reconstruct the encoding of *individual* reviews. As a result, it may not be able to generate meaningful text when the number of reviews is large. Furthermore, autoencoders are constrained to use simple decoders lacking attention (Bahdanau et al., 2014) and copy (Vinyals et al., 2015) mechanisms which have proven useful in the supervised setting leading to the generation of informative and detailed summaries. Problematically, a powerful decoder might be detrimental to the reconstruction objective, learning to express arbitrary distributions of the output sequence while ignoring the encoded input (Kingma and Welling, 2014; Bowman et al., 2016).

In this paper, we enable the use of supervised techniques for unsupervised summarization. Specifically, we automatically generate a synthetic training dataset from a corpus of product reviews, and use this dataset to train a more powerful neural model with supervised learning. The synthetic data is created by selecting a review from the corpus, pretending it is a summary, generating multiple noisy versions thereof and treating these as *pseudo-reviews*. The latter are obtained with two noise generation functions targeting textual units of different granularity: *segment* noising introduces noise at the word- and phrase-level, while *document* noising replaces a review with a semantically similar one. We use the synthetic data to train a neural model that learns to denoise the pseudo-reviews and generate the summary. This is motivated by how humans write opinion summaries, where denoising can be seen as removing diverging information. Our proposed model consists of a multi-source encoder and a decoder equipped with an attention mechanism. Additionally, we introduce three modules: (a) explicit denoising guides how the model removes noise from the input encodings, (b) partial copy enables to copy information from the source reviews only when necessary, and (c) a discriminator helps the decoder generate topically consistent text.

We perform experiments on two review datasets representing different domains (movies vs businesses) and summarization requirements (short vs longer summaries). Results based on automatic and human evaluation show that our method outperforms previous unsupervised summarization models, including the state-of-the-art abstractive system of Chu and Liu (2019) and is on the same par with a state-of-the-art supervised model (Wang and Ling, 2016) trained on a small sample of (genuine) review-summary pairs.

2 Related Work

Most previous work on unsupervised opinion summarization has focused on extractive approaches (Carenini et al., 2006; Ku et al., 2006; Paul et al., 2010; Angelidis and Lapata, 2018) where a clustering model groups opinions of the same aspect, and a sentence extraction model identifies text representative of each cluster. Ganesan et al. (2010) propose a graph-based abstractive framework for generating concise opinion summaries, while Di Fabbrizio et al. (2014) use an extractive system to first select salient sentences and then generate an abstractive summary based on hand-written templates (Carenini and Moore, 2006).

As mentioned earlier, we follow the setting of Chu and Liu (2019) in assuming that we have access to reviews but no gold-standard summaries. Their model learns to generate opinion summaries by reconstructing a canonical review of the average encoding of input reviews. Our proposed method is also abstractive and neural-based, but eschews the use of an autoencoder in favor of supervised sequence-to-sequence learning through the creation of a synthetic training dataset. Concurrently with our work, Bražinskas et al. (2019) use a hierarchical variational autoencoder to learn a latent code of the summary. While they also use randomly sampled reviews for supervised training, our dataset construction method is more principled making use of linguistically motivated noise functions.

Our work relates to denoising autoencoders (DAEs; Vincent et al., 2008), which have been effectively used as unsupervised methods for various NLP tasks. Earlier approaches have shown that DAEs can be used to learn high-level text representations for domain adaptation (Glorot et al., 2011) and multimodal representations of textual and visual input (Silberer and Lapata, 2014). Recent

work has applied DAEs to text generation tasks, specifically to data-to-text generation (Freitag and Roy, 2018) and extractive sentence compression (Fevry and Phang, 2018). Our model differs from these approaches in two respects. Firstly, while previous work has adopted trivial noising methods such as randomly adding or removing words (Fevry and Phang, 2018) and randomly corrupting encodings (Silberer and Lapata, 2014), our noise generators are more linguistically informed and suitable for the opinion summarization task. Secondly, while in Freitag and Roy (2018) the decoder is limited to vanilla RNNs, our noising method enables the use of more complex architectures, enhanced with attention and copy mechanisms, which are known to improve the performance of summarization systems (Rush et al., 2015; See et al., 2017).

3 Modeling Approach

Let $\mathbf{X} = \{x_1, ..., x_N\}$ denote a set of reviews about a product (e.g., a movie or business). Our aim is to generate a summary y of the opinions expressed in $\mathbf{X}$. We further assume access to a corpus $\mathbb{C} = \{\mathbf{X}_1, ..., \mathbf{X}_M\}$ containing multiple reviews about M products without corresponding opinion summaries.

Our method consists of two parts. We first create a synthetic dataset $\mathbb{D} = \{(\mathbf{X}, y)\}$ consisting of summary-review pairs. Specifically, we sample review x_i from $\mathbb{C}$, pretend it is a summary, and generate multiple noisy versions thereof (i.e., pseudo-reviews). At training time, a denoising model learns to remove the noise from the reviews and generate the summary. At test time, the same denoising model is used to summarize actual reviews. We use denoising as an auxiliary task for opinion summarization to simulate the fact that summaries tend to omit opinions that do not represent consensus (i.e., noise in the pseudo-review), but include salient opinions found in most reviews (i.e., non-noisy parts of the pseudo-review).

3.1 Synthetic Dataset Creation via Noising

We sample a review as a candidate summary and generate noisy versions thereof, using two functions: (a) segment noising adds noise at the token and chunk level, and (b) document noising adds noise at the text level. The noise functions are illustrated in Figure 1.

Summary Sampling Summaries and reviews follow different writing conventions. For example, reviews are subjective, and often include first-person singular pronouns such as *I* and *my* and several unnecessary characters or symbols. They may also vary in length and detail. We discard reviews from corpus $\mathbb{C}$ which display an excess of these characteristics based on a list of domain-specific constraints (detailed in Section 4). We sample a review y from the filtered corpus, which we use as the candidate summary.

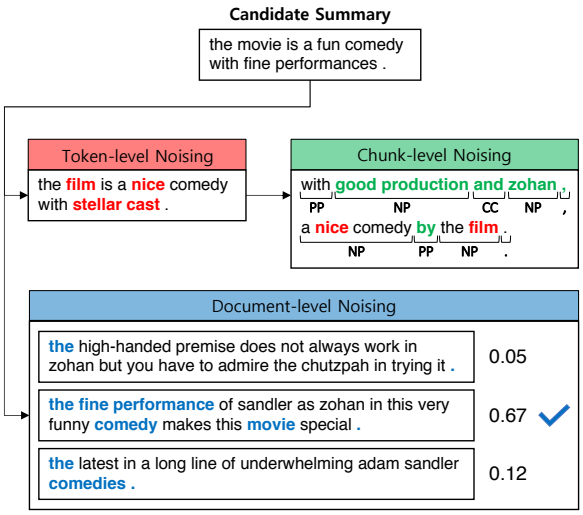

Figure 1: Synthetic dataset creation. Given a sampled candidate summary, we add noise using two methods: (a) segment noising performs token- and chunk-level alterations, and (b) document noising replaces the text with a semantically similar review.

Segment Noising Given candidate summary $y = \{w_1, ..., w_L\}$, we create a set of segment-level noisy versions $\mathbf{X}^{(c)} = \{x_1^{(c)}, ..., x_N^{(c)}\}$. Previous work has adopted noising techniques based on random n-gram alterations (Fevry and Phang, 2018), however, we instead rely on two simple, linguistically informed noise functions. Firstly, we train a bidirectional language model (BiLM; Peters et al., 2018) on the review corpus $\mathbb{C}$. For each word in y, the BiLM predicts a softmax word distribution which can be used to replace words. Secondly, we utilize FLAIR[1] (Akbik et al., 2019), an off-the-shelf state-of-the-art syntactic chunker that leverages contextual embeddings, to shallow parse each review r in corpus $\mathbb{C}$. This results in a list of chunks $\mathbf{C}_r = \{c_1, ..., c_K\}$ with corresponding syntactic labels $\mathbf{G}_r = \{g_1, ..., g_K\}$ for each review r, which we use for replacing and rearranging chunks.

Segment-level noise involves token- and chunk-

[1] https://github.com/zalandoresearch/flair

level alterations. Token-level alterations are performed by replacing tokens in y with probability $p^{\mathcal{R}}$. Specifically, we replace token w_j in y, by sampling token w'_j from the BiLM predicted word distribution (see in Figure 1). We use nucleus sampling (Holtzman et al., 2019), which samples from a rescaled distribution of words with probability higher than a threshold $p^{\mathcal{N}}$, instead of the original distribution. This has been shown to yield better samples in comparison to top-k sampling, mitigating the problem of text degeneration (Holtzman et al., 2019).

Chunk-level alterations are performed by removing and inserting chunks in y, and rearranging them based on a sampled syntactic template. Specifically, we first shallow parse y using FLAIR, obtaining a list of chunks $\mathbf{C}_y$, each of which is removed with probability $p^{\mathcal{R}}$. We then randomly sample a review r from our corpus and use its sequence of chunk labels $\mathbf{G}_r$ as a syntactic template, which we fill in with chunks in $\mathbf{C}_y$ (sampled without replacement), if available, or with chunks in corpus $\mathbb{C}$, otherwise. This results in a noisy version $x^{(c)}$ (see Figure 1 for an example). Repeating the process N times produces the noisy set $\mathbf{X}^{(c)}$. We describe this process step-by-step in the Appendix.

Document Noising Given candidate summary $y = \{w_1, ..., w_L\}$, we also create another set of document-level noisy versions $\mathbf{X}^{(d)} = \{x_1^{(d)}, ..., x_N^{(d)}\}$. Instead of manipulating parts of the summary, we altogether replace it with a similar review from the corpus and treat it as a noisy version. Specifically, we select N reviews that are most similar to y and discuss the same product. To measure similarity, we use IDF-weighted ROUGE-1 F1 (Lin, 2004), where we calculate the lexical overlap between the review and the candidate summary, weighted by token importance:

$$overlap = \sum_{w_j \in x} \big(\text{IDF}(w_j) * 1(w_j \in y)\big)$$

$$\text{P} = overlap/|x| \qquad \text{R} = overlap/|y|$$

$$\text{F}_1 = (2 * \text{P} * \text{R})/(\text{P} + \text{R})$$

where x is a review in the corpus, $1(\cdot)$ is an indicator function, and P, R, and F_1 are the ROUGE-1 precision, recall, and F_1, respectively. The reviews with the highest F_1 are selected as noisy versions of y, resulting in the noisy set $\mathbf{X}^{(d)}$ (see Figure 1).

We create a total of $2 * N$ noisy versions of y, i.e., $\mathbf{X} = \mathbf{X}^{(c)} \cup \mathbf{X}^{(d)}$ and obtain our synthetic training data $\mathbb{D} = \{(\mathbf{X}, y)\}$ by generating $|\mathbb{D}|$ pseudo-review-summary pairs. Both noising methods are necessary to achieve aspect diversity amongst input reviews. Segment noising creates reviews which may mention aspects not found in the summary, while document noising creates reviews with content similar to the summary. Relying on either noise function alone decreases performance (see the ablation studies in Section 5). We show examples of these noisy versions in the Appendix.

3.2 Summarization via Denoising

We summarize (aka denoise) the input $\mathbf{X}$ with our model which we call DENOISESUM, illustrated in Figure 2. A multi-source encoder produces an encoding for each pseudo-review. The encodings are further corrected via an explicit denoising module, and then fused into an aggregate encoding for each type of noise. Finally, the fused encodings are passed to a decoder with a partial copy mechanism to generate the summary y.

Multi-Source Encoder For each pseudo-review $x_j \in \mathbf{X}$ where $x_j = \{w_1, ..., w_L\}$ and w_k is the kth token in x_j, we obtain contextualized token encodings $\{h_k\}$ and an overall review encoding d_j with a BiLSTM encoder (Hochreiter and Schmidhuber, 1997):

$$\overrightarrow{h}_k = \text{LSTM}_f(w_k, \overrightarrow{h}_{k-1})$$
$$\overleftarrow{h}_k = \text{LSTM}_b(w_k, \overleftarrow{h}_{k+1})$$
$$h_k = [\overrightarrow{h}_k; \overleftarrow{h}_k]$$
$$d_j = [\overrightarrow{h}_L; \overleftarrow{h}_1]$$

where $\overrightarrow{h}_k$ and $\overleftarrow{h}_k$ are forward and backward hidden states of the BiLSTM at timestep k, and ; denotes concatenation (see module (a) in Figure 2).

Explicit Denoising The model should be able to remove noise from the encodings before decoding the text. While previous methods (Vincent et al., 2008; Freitag and Roy, 2018) *implicitly* assign the denoising task to the encoder, we propose an *explicit* denoising component (see module (b) in Figure 2). Specifically, we create a correction vector $c_j^{(c)}$ for each pseudo-review $d_j^{(c)}$ which resulted from the application of segment noise. $c_j^{(c)}$ represents the adjustment needed to denoise each dimension of $d_j^{(c)}$ and is used to create $\hat{d}_j^{(c)}$, a denoised

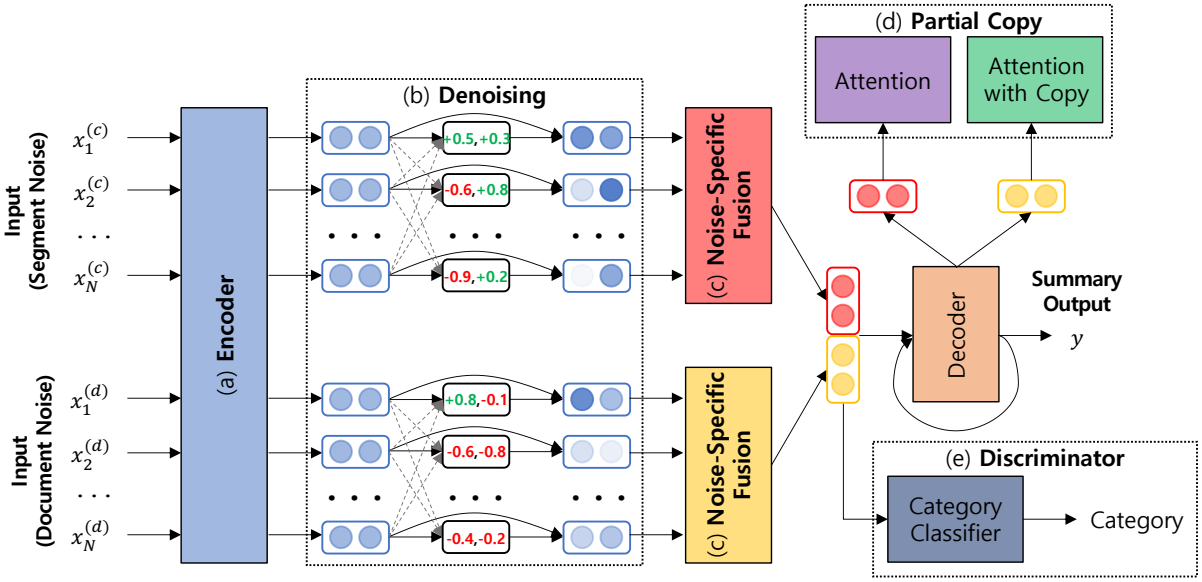

Figure 2: Architecture of DENOISESUM: it consists of a multi-source encoder with explicit denoising, noise-specific fusion, a decoder with partial copy, and a review category classifier.

encoding of $d_j^{(c)}$:

$$q = \sum_{j=1}^{N} d_j^{(c)}/N$$
$$c_j^{(c)} = \tanh(W_d^{(c)}[d_j^{(c)}; q] + b_d^{(c)})$$
$$\hat{d}_j^{(c)} = d_j^{(c)} + c_j^{(c)}$$

where q represents a mean review encoding and functions as a query vector, W and b are learned parameters, and superscript $^{(c)}$ signifies segment noising. We can interpret the correction vector as removing or adding information to each dimension when its value is negative or positive, respectively. Analogously, we obtain $\hat{d}_j^{(d)}$ for pseudo-reviews $d_j^{(d)}$ which have been created with document noising.

Noise-Specific Fusion For each type of noise (segment and document), we create a noise-specific aggregate encoding by fusing the denoised encodings into one (see module (c) in Figure 2). Given $\{\hat{d}_j^{(c)}\}$, the set of denoised encodings corresponding to segment noisy inputs, we create aggregate encoding $s_0^{(c)}$:

$$\alpha_j^{(c)} = \text{softmax}(W_f^{(c)}\hat{d}_j^{(c)} + b_f^{(c)})$$
$$s_0^{(c)} = \sum_j \hat{d}_j^{(c)} * \alpha_j^{(c)}$$

where α_j is a gate vector with the same dimensionality as the denoised encodings. Analogously, we obtain $s_0^{(d)}$ from the denoised encodings $\{\hat{d}_j^{(d)}\}$ corresponding to document noisy inputs.

Decoder with Partial Copy Our decoder generates a summary given encodings $s_0^{(c)}$ and $s_0^{(d)}$ as input. An advantage of our method is its ability to incorporate techniques used in supervised models, such as attention (Bahdanau et al., 2014) and copy (Vinyals et al., 2015). Pseudo-reviews created using segment noising include various chunk permutations, which could result to ungrammatical and incoherent text. Using a copy mechanism on these texts may hurt the fluency of the output. We therefore allow copy on document noisy inputs only (see module (d) in Figure 2).

We use two LSTM decoders for the aggregate encodings, one equipped with attention and copy mechanisms, and one without copy mechanism. We then combine the results of these decoders using a learned gate. Specifically, token w_t at timestep t is predicted as:

$$s_t^{(c)}, p^{(c)}(w_t) = \text{LSTM}_{\text{att}}(w_{t-1}, s_{t-1}^{(c)})$$
$$s_t^{(d)}, p^{(d)}(w_t) = \text{LSTM}_{\text{att+copy}}(w_{t-1}, s_{t-1}^{(d)})$$
$$\lambda_t = \sigma(W_p[w_{t-1}; s_t^{(c)}; s_t^{(d)}] + b_p)$$
$$p(w_t) = \lambda_t * p^{(c)}(w_t) + (1 - \lambda_t) * p^{(d)}(w_t)$$

where s_t and $p(w_t)$ are the hidden state and predicted token distribution at timestep t, and $\sigma(\cdot)$ is the sigmoid function.

3.3 Training and Inference

We use a maximum likelihood loss to optimize the generation probability distribution based on summary $y = \{w_1, ..., w_L\}$ from our synthetic dataset:

$$\mathcal{L}_{gen} = -\sum_{w_t \in y} \log p(w_t)$$

The decoder depends on $\mathcal{L}_{gen}$ to generate meaningful, denoised outputs. As this is a rather *indirect* way to optimize our denoising module, we additionally use a discriminative loss providing *direct* supervision. The discriminator operates at the output of the fusion module and predicts the category distribution $p(z)$ of the output summary y (see module (e) in Figure 2). The type of categories varies across domains. For movies, categories can be information about their genre (e.g., drama, comedy), while for businesses their specific type (e.g., restaurant, beauty parlor). This information is often included in reviews but we assume otherwise and use an LDA topic model (Blei et al., 2003) to infer $p(z)$ (we present experiments with human labeled and automatically induced categories in Section 5). An MLP classifier takes as input aggregate encodings $s^{(c)}$ and $s^{(d)}$ and infers $q(z)$. The discriminator is trained by calculating the KL divergence between predicted and actual category distributions $q(z)$ and $p(z)$:

$$q(z) = \text{MLP}_d(s^{(c)}, s^{(d)})$$
$$\mathcal{L}_{disc} = D_{KL}(p(z) \parallel q(z))$$

The final objective is the sum of both loss functions:

$$\mathcal{L} = \mathcal{L}_{gen} + \mathcal{L}_{disc}$$

At test time, we are given genuine reviews $\mathbf{X}$ as input instead of the synthetic ones. We generate a summary by treating $\mathbf{X}$ as $\mathbf{X}^{(c)}$ and $\mathbf{X}^{(d)}$, i.e., the outcome of segment and document noising.

4 Experimental Setup

Dataset We performed experiments on two datasets which represent different domains and summary types. The Rotten Tomatoes dataset[2] (Wang and Ling, 2016) contains a large set of reviews for various movies written by critics. Each set of reviews has a gold-standard consensus summary written by an editor. We follow the partition of Wang and Ling (2016) but do not use ground truth summaries during training to simulate our unsupervised setting. The Yelp dataset[3] in Chu and Liu (2019) includes a large training corpus of reviews without gold-standard summaries. The latter are provided for the development and test set and were generated by an Amazon Mechanical Turker. We follow the splits introduced in their work. A comparison between the two datasets is provided in Table 1. As can be seen, Rotten Tomatoes summaries are generally short, while Yelp reviews are three times longer. Interestingly, there are a lot more reviews to summarize in Rotten Tomatoes (approximately 100 reviews) while input reviews in Yelp are considerably less (i.e., 8 reviews).

Rotten Tomatoes	Train*	Dev	Test
#movies	25k	536	737
#reviews/movie	40.0	98.0	100.3
#tokens/review	28.4	23.5	23.6
#tokens/summary	22.7	23.6	23.8
corpus size			245,848

Yelp	Train*	Dev	Test
#businesses	100k	100	100
#reviews/business	8.0	8.0	8.0
#tokens/review	72.3	70.3	67.8
#tokens/summary	64.8	70.9	67.3
corpus size			2,320,800

Table 1: Dataset statistics; Train* column refers to the synthetic data we created through noising (Section 3.1).

Implementation To create the synthetic dataset, we sample candidate summaries using the following constraints: (1) the number of non-alphanumeric symbols must be less than 3, (2) there must be no first-person singular pronouns (not used for Yelp), and (3) the number of tokens must be between 20 to 30 (50 to 90 for Yelp). We set $p^{\mathcal{R}}$ to 0.8 and 0.4 for token and chunk noise, and $p^{\mathcal{N}}$ to 0.9. For each review-summary pair, the number of reviews N is sampled from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ where μ and σ are the mean and standard deviation of the number of reviews in the development set. We created 25k (Rotten Tomatoes) and 100k (Yelp) pseudo-reviews for our synthetic datasets (see Table 1).

We set the dimensions of the word embeddings to 300, the vocabulary size to 50k, the hidden di-

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

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

Model	METEOR	RSU4	R1	R2	RL
ORACLE	12.10	12.01	30.94	10.75	24.95
LEXRANK*	5.59	3.98	—	—	—
WORD2VEC	6.14	4.04	13.93	2.10	10.81
SENTINEURON	7.02	4.77	15.90	2.01	11.74
OPINOSIS*	6.07	4.90	—	—	—
MEANSUM	6.07	4.41	15.79	1.94	12.26
DENOISESUM	**8.30**	**6.84**	**21.26**	**4.61**	**16.27**
Best Supervised*	8.50	7.39	21.19	7.64	17.80

Table 2: Automatic evaluation on **Rotten Tomatoes**. Results from Amplayo and Lapata (2019) are marked with an asterisk *. Extractive/abstractive models shown in the first/second block. Best performing results for unsupervised models are **boldfaced**.

Model	R1	R2	RL
ORACLE	31.07	6.11	18.11
LEXRANK	24.62	3.66	14.51
WORD2VEC*	24.61	2.85	13.81
SENTINEURON	25.05	3.09	14.56
OPINOSIS	20.85	1.52	11.46
MEANSUM*	28.86	3.66	15.91
DENOISESUM	**30.14**	**4.99**	**17.65**

Table 3: Automatic evaluation on **Yelp**. Results from Chu and Liu (2019) are marked with an asterisk *. Extractive/abstractive models shown in the first/second block. Best performing unsupervised models are **boldfaced**.

mensions to 256, the batch size to 8, and dropout (Srivastava et al., 2014) to 0.1. For our discriminator, we employed an LDA topic model trained on the review corpus, with 50 (Rotten Tomatoes) and 100 (Yelp) topics (tuned on the development set). The LSTM weights were pretrained with a language modeling objective, using the corpus as training data. For Yelp, we additionally trained a coverage mechanism (See et al., 2017) in a separate training phase to avoid repetition. We used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001 and l_2 constraint of 3. At test time, summaries were generated using length normalized beam search with a beam size of 5. We performed early stopping based on the performance of the model on the development set. Our model was trained on a single GeForce GTX 1080 Ti GPU and is implemented using PyTorch.[4]

Comparison Systems We compared DENOISESUM to several unsupervised extractive and abstractive methods. Extractive approaches include (a) LEXRANK (Erkan and Radev, 2004), an algorithm similar to PageRank that generates summaries by selecting the most salient sentences, (b) WORD2VEC (Rossiello et al., 2017), a centroid-based method which represents the input as IDF-weighted word embeddings and selects as summary the review closest to the centroid, and (c) SENTINEURON, which is similar to WORD2VEC but uses a language model called Sentiment Neuron (Radford et al., 2017) as input representation. As an upper bound, ORACLE selects as summary the review which maximizes the ROUGE-1/2/L F1 score against the gold summary.

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

Model	RT	Yelp
DENOISESUM	16.27	17.65
10% synthetic dataset	15.39	16.22
50% synthetic dataset	15.76	17.54
no segment noising	16.03	16.88
no document noising	16.22	16.67
no explicit denoising	16.06	17.06
no partial copy	15.89	16.31
no discriminator	15.84	16.64
using human categories	15.87	15.86

Table 4: ROUGE-L of our model and versions thereof with less synthetic data (second block), using only one noising method (third block), and without some modules (fourth block). A more comprehensive table and discussion can be found in the Appendix.

Abstractive methods include (d) OPINOSIS (Ganesan et al., 2010), a graph-based summarizer that generates concise summaries of highly redundant opinions, and (e) MEANSUM (Chu and Liu, 2019), a neural model that generates a summary by reconstructing text from aggregate encodings of reviews. Finally, for Rotten Tomatoes, we also compared with the state-of-the-art supervised model proposed in Amplayo and Lapata (2019) which used the original training split. Examples of system summaries are shown in the Appendix.

5 Results

Automatic Evaluation Our results on Rotten Tomatoes are shown in Table 2. Following previous work (Wang and Ling, 2016; Amplayo and Lapata, 2019) we report five metrics: METEOR (Denkowski and Lavie, 2014), a recall-oriented metric that rewards matching stems, synonyms, and

Model	RT Inf	RT Coh	RT Gram	Yelp Inf	Yelp Coh	Yelp Gram
SENTINEURON	11.8	8.3	**25.4**	-24.8	-0.8	**9.3**
MEANSUM	-32.1	-34.4	-46.8	6.3	-7.5	-10.8
DENOISESUM	**20.3**	**26.1**	21.4	**18.5**	**8.2**	1.6

Model	Yelp FullSupp	Yelp PartSupp	Yelp NoSupp
MEANSUM	41.7%	20.4%	38.0%
DENOISESUM	55.1%	24.3%	20.5%
GOLD	63.6%	23.6%	12.8%

Table 5: Best-worst scaling (left) and summary veridicality (right) evaluation. Between systems differences are all significant, using a one-way ANOVA with posthoc Tukey HSD tests ($p < 0.01$).

paraphrases; ROUGE-SU4 (Lin, 2004), the recall of unigrams and skip-bigrams of up to four words; and the F1-score of ROUGE-1/2/L, which respectively measures word-overlap, bigram-overlap, and the longest common subsequence between system and reference summaries. Results on Yelp are given in Table 3 where we compare systems using ROUGE-1/2/L F1, following Chu and Liu (2019).

As can be seen, DENOISESUM outperforms all competing models on both datasets. When compared to MEANSUM, the difference in performance is especially large on Rotten Tomatoes, where we see a 4.01 improvement in ROUGE-L. We believe this is because MEANSUM does not learn to reconstruct encodings of aggregated inputs, and as a result it is unable to produce meaningful summaries when the number of input reviews is large, as is the case for Rotten Tomatoes. In fact, the best extractive model, SENTINEURON, slightly outperforms MEANSUM on this dataset across metrics with the exception of ROUGE-L. When compared to the best supervised system, DENOISESUM performs comparably on several metrics, specifically METEOR and ROUGE-1, however there is still a gap on ROUGE-2, showing the limitations of systems trained without gold-standard summaries.

Table 4 presents various ablation studies on Rotten Tomatoes (RT) and Yelp which assess the contribution of different model components. Our experiments confirm that increasing the size of the synthetic data improves performance, and that both segment and document noising are useful. We also show that explicit denoising, partial copy, and the discriminator help achieve best results. Finally, human-labeled categories (instead of LDA topics) decrease model performance, which suggests that more useful labels can be approximated by automatic means.

Human Evaluation We also conducted two judgment elicitation studies using the Amazon Mechanical Turk (AMT) crowdsourcing platform. The first study assessed the quality of the summaries 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/business name, some basic background information, and a gold-standard summary. They were also presented with three system summaries, produced by SENTINEURON (best extractive model), MEANSUM (most related unsupervised model), and DENOISESUM.

Participants were asked to select the *best* and *worst* among system summaries taking into account how much they deviated from the ground truth summary in terms of: *Informativeness* (i.e., does the summary present opinions about specific aspects of the movie/business in a concise manner?), *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 and grammatical?). We randomly selected 50 instances from the test set. We collected five 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. Results are reported in Table 5, where Inf, Coh, and Gram are shorthands for Informativeness, Coherence, and Grammaticality. DENOISESUM was ranked best in terms of informativeness and coherence, while the extractive system SENTINEURON was ranked best on grammaticality. This is not entirely surprising since extractive summaries written by humans are by definition grammatical.

Our second study examined the veridicality of the generated summaries, namely whether the facts mentioned in them are indeed discussed in the input reviews. Participants were shown reviews and the corresponding summary and were asked to verify for each summary sentence whether it was fully supported by the reviews, partially supported, or not at all supported. We performed this experiment

on Yelp only since the number of reviews is small and participants could read them all in a timely fashion. We used the same 50 instances as in our first study and collected five judgments per instance. Participants assessed the summaries produced by MEANSUM and DENOISESUM. We also included GOLD-standard summaries as an upper bound but no output from an extractive system as it by default contains facts mentioned in the reviews.

Table 5 reports the percentage of fully (FullSupp), partially (PartSupp), and un-supported (NoSupp) sentences. Gold summaries display the highest percentage of fully supported sentences (63.3%), followed by DENOISESUM (55.1%), and MEANSUM (41.7%). These results are encouraging, indicating that our model hallucinates to a lesser extent compared to MEANSUM.

6 Conclusions

We consider an unsupervised learning setting for opinion summarization where there are only reviews available without corresponding summaries. Our key insight is to enable the use of supervised techniques by creating synthetic review-summary pairs using noise generation methods. Our summarization model, DENOISESUM, introduces explicit denoising, partial copy, and discrimination modules which improve overall summary quality, outperforming competitive systems by a wide margin. In the future, we would like to model aspects and sentiment more explicitly as well as apply some of the techniques presented here to unsupervised single-document summarization.

Acknowledgments

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

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Reinald Kim Amplayo and Mirella Lapata. 2019. Informative and controllable opinion summarization. *CoRR*, abs/1909.02322.

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.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2019. Unsupervised multi-document opinion summarization as copycat-review generation. *arXiv preprint arXiv:1911.02247*.

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

Giuseppe Carenini and Johanna D. Moore. 2006. Generating and evaluating evaluative arguments. *Artif. Intell.*, 170(11):925–952.

Giuseppe Carenini, Raymond Ng, and Adam Pauls. 2006. Multi-document summarization of evaluative text. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

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, pages 1223–1232, Long Beach, California.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the 9th Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland. 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. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Thibault Fevry and Jason Phang. 2018. Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422, Brussels, Belgium. Association for Computational Linguistics.

Markus Freitag and Scott Roy. 2018. Unsupervised natural language generation with denoising autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3922–3929, Brussels, Belgium. 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.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

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.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 513–520, Bellevue, Washington. Omnipress.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

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

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Minqing Hu and Bing Liu. 2006. Opinion extraction and summarization on the web. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, pages 1621–1624, Boston, Massachusetts. AAAI Press.

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, California.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proceedings of the 3rd International Conference on Learning Representations*, Banff, Alberta.

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.

Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 100–107, Palo Alto, California.

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, 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.

Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of the 18th International Conference on World Wide Web*, pages 131–140, Madrid, Spain. ACM.

Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, pages 339–348. Association for Computational Linguistics.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, pages 79–86. Association for Computational Linguistics.

Michael Paul, ChengXiang Zhai, and Roxana Girju. 2010. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 66–76, Cambridge, MA. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada.

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116, Florence, Italy. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. 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 *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization - Volume 4*, pages 21–30. Association for Computational Linguistics.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444.

Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. 2017. Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 12–21, Valencia, Spain. Association for Computational Linguistics.

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.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12).

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.

Carina Silberer and Mirella Lapata. 2014. Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732, Baltimore, Maryland. 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. *Journal of Machine Learning Research*, 15(1):1929–1958.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103, Helsinki, Finland.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, pages 2692–2700, Montréal, Canada.

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.