Self-Supervised and Controlled Multi-Document Opinion Summarization

Hady Elsahar,[1] **Maximin Coavoux,**[2] **Matthias Gallé,**[1] **Jos Rozen**[1]
{hady.elsahar,matthias.galle,jos.rozen}@naverlabs.com
maximin.coavoux@univ-grenoble-alpes.fr
[1]Naver Labs Europe
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG

Abstract

We address the problem of unsupervised abstractive summarization of collections of user generated reviews through self-supervision and control. We propose a self-supervised setup that considers an individual document as a target summary for a set of similar documents. This setting makes training simpler than previous approaches by relying only on standard log-likelihood loss and mainstream models. We address the problem of hallucinations through the use of control codes, to steer the generation towards more coherent and relevant summaries. Our benchmarks on two English datasets against graph-based and recent neural abstractive unsupervised models show that our proposed method generates summaries with a superior quality and relevance, as well as a high sentiment and topic alignment with the input reviews. This is confirmed in our human evaluation which focuses explicitly on the faithfulness of generated summaries. We also provide an ablation study showing the importance of the control setup in controlling hallucinations.

1 Introduction

Recent progress in unsupervised methods has created breakthroughs in natural language processing applications, such as machine translation (Artetxe et al., 2018; Lample et al., 2018). Those have been mostly based on a bootstrapping approach, which consists in iteratively alternating between two representations, and optimizing a reconstruction loss. Beyond machine translation, other applications include Question-Answering (Lewis et al., 2019) and parsing (Drozdov et al., 2019). While similar ideas have been applied as well for video summarization (Yuan et al., 2019), such a bootstrapping approach seems less suited for summarization, because of the inherent information loss when going from the full text to the summarized one. Existing unsupervised approaches for summarization therefore relied mostly on extractive graph-based systems (Mihalcea and Tarau, 2004). Only recently have there been proposals for unsupervised abstractive summarization, using auto-encoders (Chu and Liu, 2019; Bražinskas et al., 2020). However, these set-ups are complex and require a combination of loss functions (Chu and Liu, 2019) or hierarchical latent variables (Bražinskas et al., 2020) to ensure that the generated summaries remain on-topic.

In this paper, we investigate a self-supervised approach for multi-document opinion summarization. In this setting, there are multiple opinions (reviews), one entity (products, venues, movies, etc) and the goal is to extract a short summary of those opinions. Our approach is based on self-supervision and does not require any gold summaries. We train a supervised model on examples artificially created by selecting (i) one review that will act as a target summary and (ii) a subset of reviews of the same entity that acts as a document collection.

Neural models have a known problem of hallucination (Rohrbach et al., 2018), which can be misleading in natural language generation tasks as the fluency of those models often distract from the wrong facts stated in the generated text. To reduce this effect, we propose to use control tokens (Fan et al., 2018; Keskar et al., 2019; Krause et al., 2020). Control tokens are discrete variables that are used to condition the generation. Different from previous work, our goal is not to allow users to control the generated text, but instead to steer the generated text to produce an output which is consistent with the input documents to be summarized.

Our main contributions are therefore three-fold:

- performing multi-document summarization by modelling it as a self-supervised problem where one document acts as the summary of a subset. We carefully select those two, and

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1646–1662
April 19 - 23, 2021.

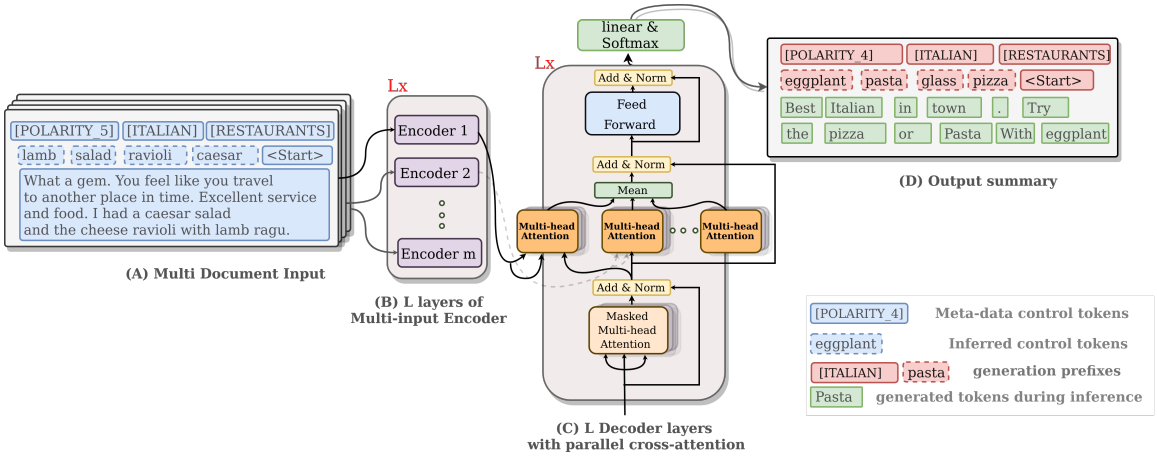

Figure 1: Description of our proposed model: (A) is the set of input reviews, augmented with control tokens (from meta-data in uppercase, inferred in lowercase). (B) is the encoder, which is run separately on each input review. The standard Transformer decoder is modified in (C) to allow for Parallel cross-attention on different inputs separately. Finally, (D) is the generated output. During inference the control tokens are fed as prompts to the decoder and generation starts afterwards.

link the resulting formulation to a recent theoretical framework (Peyrard, 2019) (Sect. 3);

- using control tokens to steer the model towards consistency, increasing relevance of the generated summary (Sect. 4);
- an application of multi-input transformer model (Libovický et al., 2018) to summarization. This model encodes each input independently, and at decoding time applies parallel attention to each encoded input (Sect. 5).

Our experimental results (Sect. 6 and 7) show that our approach outperforms existing models on two datasets: Yelp reviews on venues (Chu and Liu, 2019) and Rotten Tomatoes movie reviews (Wang and Ling, 2016). We focus the human evaluation on the faithfulness of the summaries, confirming they are more factually correct than baselines.

2 Related Work

Unsupervised Opinion Summarization Extractive summarization consists in selecting a few sentences from the input documents to form the output summary. The centroid method (Radev et al., 2004; Rossiello et al., 2017; Gholipour Ghalandari, 2017) consists in ranking sentences according to their relevance to the whole input. Graph-based methods, such as LexRank (Erkan and Radev, 2004) or TextRank (Mihalcea and Tarau, 2004; Zheng and Lapata, 2019), use the PageRank algorithm to find the most central sentences in a graph of input sentences, where edge weights indicate word overlap. In contrast to these methods, we focus on abstractive summarization methods.

Non-neural abstractive methods (Ganesan et al., 2010; Nayeem et al., 2018) are also graph-based, but work on word-type graphs. West et al. (2019) introduced a self-supervised model for sentence compression: they use an unsupervised extractive system to generate training data for a supervised sentence compressor. Their system works on single sentences whereas our end-to-end approach summarizes multiple reviews.

Recently, a few approaches for *neural* unsupervised abstractive summarization have been proposed. Chu and Liu (2019, MeanSum) introduced a summarization system based on a review autoencoder. At inference time, MeanSum encodes every review for a product to a vector, computes the centroid of reviews and uses this centroid to seed the decoder and generate a summary. However, averaging representations of statements that are sometimes contradictory tends to confuse the decoder, and might lead it to ignore the input signal. To deal with this limitation, Coavoux et al. (2019) add a clustering step to group similar reviews and to generate one sentence per such found cluster. Bražinskas et al. (2020) proposed to solve the problem of unsupervised opinion summarization with an auto-encoder with latent variables. They use latent variable for products and reviews to address the hallucination issue, while at the same time allowing it to capture information from the set of reviews on the same entity. In contrast, we argue that our self-supervised setting is simpler as it relies on training with standard models. In addition, the use of Transformer (as opposed to GRU in their case) makes it possible to apply separate attentions to each input. Probably most similar to our self-

supervised proposal is the recent work of Amplayo and Lapata (2020), in particular their document noising sub-strategy. Compared to it, our simple selection criteria of the dataset avoids any use of (domain-specific) noise generator. In addition, our use of control tokens allows to easily include existing (or inferred) meta-information. A similar approach is also used by Shapira and Levy (2020), which trains a seq2seq model by clustering reviews and using the medoid as target summary.

Another work that has recently shown the promise of self-supervision for summarization is Zhang et al. (2020a), in which masked-out sentences are predicted from the surrounding text. Our self-supervision training mechanism can be seen as a multi-document version of that.

Controlled Generation Controllable text generation has been previously investigated to apply global constraints on text generation, by directly optimizing evaluation metrics through policy gradient methods (Ranzato et al., 2016; Wu et al., 2018; Liu et al., 2017; Li et al., 2016b; Yi et al., 2018) or continuous approximation methods (Chu and Liu, 2019; Yang et al., 2018).

Other methods applied control only at inference time. Weighted decoding (Holtzman et al., 2018) was shown to be challenging, and often detrimental to fluency and coherence (See et al., 2019). Constrained beam search (Anderson et al., 2017; Hokamp and Liu, 2017; Post and Vilar, 2018) is slower, requires very large beam sizes, and does not enable soft constraints. Finally, updating the decoder hidden states (Chen et al., 2018; Dathathri et al., 2020) requires an extra training step.

Control codes have been introduced in generation as an early form of copy mechanism (Luong et al., 2015; ElSahar et al., 2018) to address the problem of rare words. They were widely adopted to steer language models towards specific features, such as aspects (Keskar et al., 2019) or structured outputs (Zellers et al., 2019).

In prior work, controlled language models rely on a predefined set of control tokens, collected manually (Keskar et al., 2019) or from dictionaries (Dathathri et al., 2020), which can lead to low domain coverage. Nabil et al. (2014) and ElSahar and El-Beltagy (2015) construct lexicons by exploiting the feature selection ability of sentiment classifiers, an approach that produces more relevant lexicons than classical topic models (e.g. LDA, Blei et al., 2003). In our work, we also rely on classifiers using the categories of reviews provided as meta-data. Without meta-data, we could have relied instead on unsupervised or weakly supervised aspect extractors (He et al., 2017; Angelidis and Lapata, 2018).

Hierarchical encoding. In order to allow a neural summarizer to read several sections, Cohan et al. (2018) proposes a hierarchical LSTM that works at two level. Similar to our proposal, Liu and Lapata (2019) extends a Transformer network to read several ranked paragraphs as input, avoiding a retrieve-then-read pipeline. In multi-document summarization, the paragraphs are not ranked but independent. This entails a significant change model-wise. We propose to encode each review independently (avoiding inter-paragraph self-attention) and only adapt the decoder-encoder attention.

3 Self-Supervision

In order to create our training dataset we assume that a review s_i for an entity (venue or product) can serve as a summary for a set of other similar reviews D_i. This simple intuition allows us to create training points (D_i, s_i) in a very similar way to what the model will experience at inference time. However, there are two issues with this approach. First, the potential set of training points is too large to be explored exhaustively. Given the set of all reviews $\mathcal{D}$ the total number of possible input-output pairs is $2^{|\mathcal{D}|-1} \times |\mathcal{D}|$. Second, the assumption that any review is fit to serve as a summary for any set of other reviews is obviously not true, and might yield a very noisy training dataset.

To solve the combinatorial explosion, we limit the size of D_i to k, and from a given s_i, we look for a set of k *good* reviews D_i, for which s_i serves as a good summary. Fixing k also simplifies training, and enables comparison with previous work where the number of input reviews is fixed (Chu and Liu, 2019; Bražinskas et al., 2020). Both s_i and all members of D_i are reviews of the same entity.

Having s_i fixed, we now search for reviews $d_1, \ldots, d_k$ for which s_i is a relevant review:

$$\begin{aligned} \textit{rel}(s_i) &= \{d_1, d_2, ..., d_k\}, \\ &= \underset{D_i \subset \mathcal{D} \setminus \{s_i\}, |D_i|=k}{\arg\max} \sum_{d_j \in D_i} \text{sim}(s_i, d_j) \end{aligned} \tag{1}$$

where *sim* is an arbitrary similarity function (that we define at the end of this section).

Fixing first the target summaries turns traditional approaches upside down. In particular, a recently

proposed theoretical model of importance in summarization (Peyrard, 2019) defines the importance of a summary based on three aspects: (i) minimum redundancy, (ii) maximum relevance with the input document, and (iii) maximum informativeness. In that line of work D_i is considered fixed: redundancy and informativeness are not dependent on D_i and can therefore be ignored when s_i is fixed. In this setting Peyrard (2019) reduces then to Eq. 1

Then, we sort the data-points $(d_i, \textit{rel}(d_i))$ according to the value of the relevance $(\sum_{d_j \in \text{rel}(d_i)} \text{sim}(d_i, d_j))$. Depending on the desired size of the target dataset, we keep the top-T pairs for training. Limiting T inherently increases informativeness, since it limits the creation of training examples where input and outputs are repetitive similar reviews that might be very prominent on corpora level (e.g. "Great restaurant."). This method is simple and fast, thanks to nearest neighbour search libraries (Pedregosa et al., 2011b). For all our experiments we defined *sim* to be the cosine similarity over a tf-idf bag-of-word representation (Ramos et al., 2003).

4 Controlling Hallucinations

Hallucinations are pieces of generated text that bear no relationship to the text they were conditioned on. They are likely to happen in our self-supervised setting, due to the noise from the construction of training instances. This might happen if the synthetically created training data contains contradictory signals, or because certain types of review are overly present (e.g. "great movie"). The model might default to those frequent patterns if it finds itself in a unfrequent state during decoding. To alleviate the problem of hallucinations, we propose to use *control tokens* that represent desired traits of the output text to steer the generated text towards more input-coherent summaries. These control tokens are inferred from each review, and used as prompts at inference time. We use two types of codes as follows:

1) Metadata control tokens. Those are special tokens that are associated with each input review, and are the capitalized control tokens in Fig. 1. We use two types of metadata that represent (i) the review **polarity**, a numerical value denoting the average sentiment score of the input reviews; (ii) **categorical tokens** representing the type of the entity of the review (e.g. Deli, Beauty&Spa, Furniture Stores). When meta-data labels are unavailable for all reviews (as in the Rotten-Tomatoes dataset), we infer control tokens with the same process, but using categories predicted by a classifier trained on labeled examples from the same domain.

2) Inferred control tokens. We follow recent work (Keskar et al., 2019; Dathathri et al., 2020) that shows that it is preferable to condition NLG models on control tokens that naturally co-occur in text. On one side, this allows for better control, and at the same it seems to be more robust when new (previously unseen) control codes are used. Here, we propose to use control codes that represent informative aspects (e.g. wine, service, ingredients) that occur in the input reviews text. However, instead of relying on manually created bag of control tokens for each desired attribute – which comes with obvious domain coverage limitations – we propose to infer those control codes from the text corpus.

To do so, we rely on the intrinsic feature selection capabilities of regularized linear classification models. For each category ℓ in the meta-data associated with each review we train a linear support vector machine (SVM) classifier (Vapnik and Lerner, 1963)[1] that learns to classify between reviews from this category and negative examples sampled randomly from the rest of the corpus. The features of the SVMs are parameterized by the weight vector $\theta_\ell \in \mathcal{R}^d$, where d is the number of features (in our experiments: all unigrams and bigrams present in the corpus). We used a squared hinge loss with $L1$ regularization over θ_ℓ – the latter to increase sparsity and force feature selection (Tibshirani, 1996; Ng, 2004). Finally, we trim the feature list into those who correspond to positive weights and renormalize the weights. The output of this step is a ranked list of n-grams that represent the distinctive aspects of each category.

When creating training data for summarization, we enrich each review with the top weighted n-grams of their corresponding categories as follows. For a given review d about entity p, we consider all m labels of p and use the weights of the corresponding classifiers $\theta_{\ell_i^{(p)}}$ (for each label $\ell_i^{(p)}$ of p). We only consider those n-grams actually occurring in d, and keep the top 8 such features. Note that these features could come from different classifiers, as we consider all m labels.

During training, we enrich each review with its tailored control codes. In particular, the reviews acting as summary also contain them, and by con-

[1] We use `liblinear` (Fan et al., 2008).

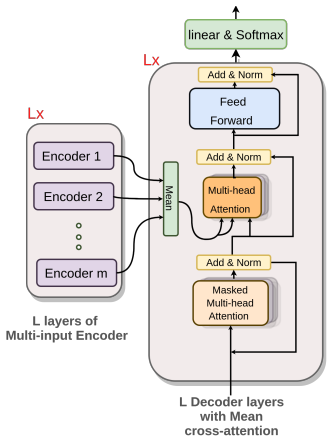

Figure 2: Our adaptation of the Transformer cross-attention to allow *Mean* combination of multi-sources.

struction those are n-grams present in the text. At inference time – when the target side and its control codes are not available – we select the most repeated control tokens from the input side and feed them as a prefix to the decoder. There is clearly a risk that the model just learns to copy the control codes it has seen somewhere in the text. We check whether this is the case in Sect. 7.

5 Multi-source Transformer Model

Previous work for multi-document summarization (Chu and Liu, 2019) built multi-source input representations through a simple mean over the last hidden states of the encoder. An intrinsic limitation of this method is that the full set of reviews is represented as a single vector. This aggregation might cause information distortion especially when some input reviews are expected to have conflicted opinions in between. Standard transformer models (Vaswani et al., 2017) consider only a single input to the decoder part of the model. Aggregating all input reviews into a single input (Junczys-Dowmunt, 2019) with special tokens to represent document boundaries might be slow and impractical due the $O(n^2)$ complexity of the self-attention mechanism. We therefore experiment with several input combination strategies of the transformer cross-attention (Libovický et al., 2018).

Parallel. At each cross-attention head, the decoder set of queries Q attend to each of the encoded inputs separately from which the set of keys ($K_i \in K_{1:m}$) and values ($V_i \in V_{1:m}$) are generated and then the yielded context is averaged and followed by a residual connection from the previous decoder layer (box C in Fig. 1).

$$A^h_{\text{parallel}}(Q, K_{1:m}, V_{1:m}) = \frac{1}{m}\sum_{i=1}^{m} A^h(Q, K_i, V_i).$$

Mean. We also propose a simpler and less computationally demanding input combination strategy. It does not apply the cross-attention with each encoder separately. Instead, the set of keys and values coming from each input encoder are aggregated using the average at each absolute position. Afterwards the decoder set of queries attend to this aggregated set of keys and values. This combination can be seen as a more efficient variation of the flat combination strategy (Libovický et al., 2018) with mean instead of concatenation. Fig. 2 depicts this strategy, which replaces box (C) in Fig. 1.

$$A^h_{\text{mean}}(Q, K_{1:m}, V_{1:m}) = A^h\left(Q, \frac{1}{|m|}\textstyle\sum_{i=1}^{m} K_i, \frac{1}{|m|}\sum_{i=1}^{m} V_i\right)$$

In practice, we share the parameters across all encoders, this can be also seen as a single encoder used to encode each input document independently. We believe that this is an appropriate design choice as the order of the input document doesn't matter. Furthermore, this is necessary to allow variable number of input documents during different training batches or during inference. In Sect. 7, we compare both approaches through an ablation study, focusing on summary quality as well as empirical training times.

6 Experimental Setup

Experimental Details All our models are implemented with PyTorch (Paszke et al., 2019) and Fairseq (Ott et al., 2019) libraries, as well as scikit-learn (Pedregosa et al., 2011a) for the classifiers used either for inferring control tokens or for evaluation. For all our models we use sentence piece (Kudo and Richardson, 2018) as a tokenizer with a vocabulary size of 32 000. We use the same hyperparameters as the Transformer Big model described by Vaswani et al. (2017) ($d_{\text{model}} = 1024$, $n_{\text{heads}} = 16$, $n_{\text{layer}} = 6$, dropout $= 0.1$). We optimize them with a Nesterov accelerated SGD optimizer with a learning rate of 0.01. We train all models for a total of 80 000 steps across 25 epochs, with linear warm-up for the first 8 000 steps. We select the best model checkpoint based on perplexity on the validation set. All models were trained on one machine with 4 NVIDIA V100 GPUs, the

	Model	ROUGE-1	ROUGE-2	ROUGE-L	F_{BERT}	Sentiment Acc.	$F_{category}$
YELP	Textrank (Mihalcea and Tarau, 2004)	28.3	4.2	14.9	84.1	82.0	53.4
	Lexrank (Radev et al., 2004)	27.4	3.9	14.9	84.2	83.5	54.1
	Opinosis (Ganesan et al., 2010)	26.8	3.4	14.2	81.2	80.5	53.0
	H-VAE (Bražinskas et al., 2020)	29.5	5.3	18.1	–	–	–
	DenoiseSum (Amplayo and Lapata, 2020)	30.1	4.9	17.6	–	–	–
	MeanSum (Chu and Liu, 2019)	28.6	3.8	15.9	86.5	83.5	50.3
	Ours	**32.8**	**8.7**	**18.8**	**86.8**	**83.9**	**55.2**
RT	Textrank	19.0	4.3	19.4	**85.3**	**75.8**	41.6
	Lexrank	17.6	3.5	18.2	**85.3**	73.2	40.9
	Opinosis	15.2	2.9	16.9	84.1	67.5	37.1
	Ours	**20.9**	**4.5**	**22.7**	**85.3**	70.9	**43.6**
	DenoiseSum*	21.2	4.6	16.27	–	–	–

Table 1: Automatic evaluations results against gold summaries of Yelp and Rotten Tomatoes (RT) datasets. "Ours" denotes our proposed system with parallel input combination strategy and control codes. In the RT experiments, we report numbers from (Amplayo and Lapata, 2020) denoted as DenoiseSum* which are not comparable as they utilize different train/dev/test splits.

	Model	Dist-1	Dist-2	Dist-3	$Dist_c$-1	$Dist_c$-2	$Dist_c$-3
Extract.	Textrank	0.68	0.95	0.992	0.135	0.62	0.90
	Lextrank	0.70	0.96	0.994	0.144	0.6	0.92
	Opinosis	0.72	0.94	0.97	**0.159**	**0.66**	**0.92**
Abstr.	MeanSum	0.72	0.95	0.98	0.091	0.39	0.67
	Ours	**0.79**	**0.99**	**1.00**	0.097	0.41	0.64

Table 2: Referenceless evaluation results on Yelp dataset.

longest model took 50 hours to train. For inference, we use a beam size of 35. We discard hypotheses that contain twice the same trigram. We limit generation of each summary to a maximum budget of 150 tokens for each summary for Yelp, as was done by Chu and Liu (2019), and a budget of 50 tokens for Rotten Tomatoes. We set a similar budget for all other extractive baselines in the experiments. Finally, we use length normalization (Wu et al., 2016) with length penalty 1.2 to account for the model's bias towards shorter sequences.

Datasets We evaluate our proposal on two English datasets: Yelp[2] (Chu and Liu, 2019) and Rotten Tomatoes (Wang and Ling, 2016). The Yelp dataset contains reviews of businesses (around 1M reviews for 40k venues). As described in Sect. 3, for each venue, we select the best reviews to use as target summaries: either the top-p (with $p = 15\%$) or the top-T (with $T = 100$) reviews, whichever is smaller. For each selected target summary, we take its $k = 8$ most similar reviews (cosine) to form its input. We obtain around 340k training examples, representing 22.5k venues. The Rotten Tomatoes dataset was constructed by (Wang and Ling, 2016) from the movie review website rottentomatoes.com. We use the same process as for Yelp, but use $p = 1\%$ and $T = 150$. We construct around 170k training examples, representing 3.7k movies. We provide more details in the supplementary material.

Evaluation Metrics We evaluate summary systems with the classical ROUGE-F-{1,2,L} metrics (Lin, 2004).[3] We also report BERT-score (Zhang et al., 2020b), a metric that uses pre-trained BERT (Devlin et al., 2019) to compute the semantic similarity between a candidate summary and the gold summary. Dist-n and $Dist_c$-n ($n = 1, 2, 3$) scores (Li et al., 2016a) are the percentage of distinct n-grams in the generated text on the summary level or the corpora level respectively. Dist-n is an indicator of repetitiveness within a single summary while $Dist_c$-n indicates the diversity of different generations. Finally, as done by Chu and Liu (2019), we use a classifier to check whether the sentiment of the summary is consistent with that of input reviews (Sentiment Acc., Tab. 1).[4] We extend this method to check whether the correct product category can also be inferred from the summary, we report $\mathbf{F}_{\text{category}}$ the micro F-score of the multi-label category classifier.

Baselines and Other Systems We compare our system to three unsupervised baselines. TextRank (Mihalcea and Tarau, 2004) and LexRank (Radev et al., 2004) are extractive systems. Opinosis (Ganesan et al., 2010) is an abstractive graph-based system. We use openly available Python implementations for TextRank[5] (Barrios et al., 2015)

[2] https://www.yelp.com/dataset/challenge

[3] For Yelp we use Chu and Liu (2019)'s implementation to keep results comparable. For RottenTomatoes we use *py-rouge* package pypi.python.org/pypi/pyrouge/0.1.3

[4] We use 3 classes: negative (1, 2 star), neutral (3), positive (4, 5). Numbers are not comparable with Chu and Liu (2019).

[5] https://github.com/summanlp/textrank

Model	*Quality* ROUGE-1	ROUGE-2	ROUGE-L	F_{BERT}	Sentiment Acc.	$F_{category}$	*Speed* Train. (wps)
$Ours_{Parallel}$	**32.8**	**8.7**	**18.8**	86.8	**83.9**	55.2	3785
$Ours_{Mean}$	29.4	5.3	17.2	**87.6**	83.4	**56.2**	8075
$Ours_{Parallel}$ − cntrl.	25.3	3.7	15.5	85.2	76.9	43.9	7609
$Ours_{Mean}$ − cntrl.	27.5	5.3	17.1	87.3	80.0	52.1	**8714**

Table 3: Ablation study showing the effectiveness of parallel-cross attention and control tokens on Yelp dataset. "−cntrl." denotes models trained without the control step. "Train. (wps)" denotes the word per second rate at training time.

OURS:
This was my *first visit* to *Capriotti's* and I really enjoyed it . I had the *Capastrami* and my husband had *the Bobbie* . We both enjoyed our *sandwiches* as well . The *quality* of the *ingredients*, however, was not what we expected . We also enjoyed the *cheese steak* as well as the *turkey*, which was not bad at all . This place is a bit on the expensive side for what you get, but you get what you pay for . The *seating* is limited, so it's a good place to visit if you're in a hurry.

MeanSum:
Drove by here for the *first time*. I just went to the *deli* with a friend and it's a quick fix that is just about as good as it gets. But it's not an actual *sandwich*, but it's not as good as I remembered it, but they were great!! *Sandwich* was also very good, just a hint of cinnamon. I will be back for the other locations.

TextRank (Extractive):
Will not return This place is always good, I think the owner actually made my sandwich last time I was there , owner or manager, anyway it was superb! Ordered a sandwich, watched the guy write it down and 25 minutes later the same person asked what I wanted when I reminded him of my sandwich, he only said he can't remember where the order went. I watched 4 people come in after me order, one person the same sandwich just a different size then me get their food, pay and leave. At that point I gave up because as much as I like their sandwiches I am never going back.

Figure 3: Examples of different model generations to the same input set of documents. Green (italics) denotes substrings with exact match with the input, red (underlined) denotes statements without support in the input. TextRank is shown as a reference: all substrings are present in the input, but note the lack of cohesion.

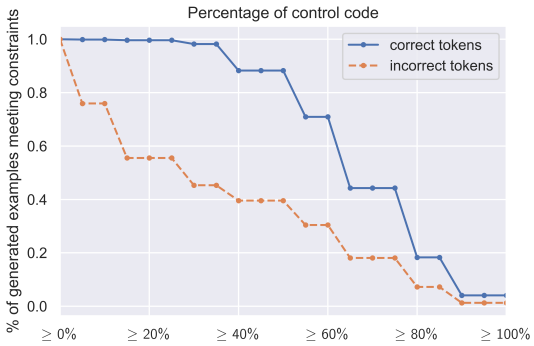

Figure 4: Proportion of control tokens fed as prompts that occur in the generated summary. When the model is fed control tokens that occur in the input reviews (*correct*) it tends to generate them in the output. Contrary to this, *incorrect* control tokens are mostly ignored.

and LexRank,[6] with their default parameters. For Opinosis, we use the official Java implementation, with default hyperparameters.[7] We also compare our systems with recent neural unsupervised summarization systems (Chu and Liu, 2019; Bražinskas et al., 2020). In addition, in our ablation study (next section) we also compare against a vanilla Transformer system, to capture the relative gains obtained on top of that model.

7 Evaluation Results

Automatic Evaluation Table 1 contains the automatic evaluation metrics with respect to reference summaries. The proposed multi-input self-supervised model with control codes perform consistently better in the Yelp dataset across the benchmarked models, including the recent neural unsupervised models of MeanSum and H-VAE. For the recent H-VAE model we report the numbers from their paper.[8] For MeanSum we re-run their provided checkpoint and run evaluation through the same pipeline. The BERTScore (Zhang et al., 2020b) differences are closer and seem to favour neural models.

With the RottenTomatoes dataset we only benchmarked the graph-based unsupervised methods, since the released pretrained MeanSum model does not cover the domain of movie reviews. We attribute the lower score in sentiment accuracy to the fact that the "summaries" in RottenTomatoes are critical reviews, written in a very different style than the original reviews.

Table 2 contains reference-less evaluation, analyzing the number of distinct n-grams (an indicator of repetitiveness) on the summary level and corpora level. On the summary level our model outperforms all the baselines: our model is capable of generating richer and less repetitive summaries. On the level of all generations our model generates

[6] https://github.com/crabcamp/lexrank

[7] Except for the redundancy parameter set to one, since the default led to many empty outputs. https://github.com/kavgan/opinosis-summarization

[8] While the ROUGE implementation might be different, the numbers of the common baselines are very close.

text with more diversity than MeanSum. In general however extractive models tend to have more diversity on the corpus level as they directly copy from each input separately, while abstractive models tend to learn repetitive patterns present in the training set.

Fig. 3 shows summaries generated by different models from the same input. We notice that our model learned to copy aspects of the input documents such as restaurant names "Capricotti's" and menu items "the Bobbie", possibly due to the cross-attention mechanism. We provide more examples as supplementary material.

Human Evaluation Existing natural language generation systems are known to generate very fluent language, that looks very natural to native speakers. On the other side, current neural models are known to generate factually incorrect data, something which was less of a concern in pre-neural methods but also much harder to detect. As mentioned by Kryscinski et al. (2019): "Neither of the methods explicitly examines the factual consistency of summaries, leaving this important dimension unchecked." Inspired by Falke et al. (2019) we decided to focus the human evaluation on those aspects of the summarization evaluation in which existing models risk failing the most, the one of *faithfulness*.

We annotated 94 summaries through a crowdsourcing platform, comparing 3 systems (Gold, MeanSum and ours). Workers were asked if "the summary contains correct information given the original reviews". In total we had 282 tasks (94×3) and each task was labeled by 3 annotators and paid \$0.50 (defined by a pilot study to aim for \$15 / hour) and restricted to experienced, English-speaking workers. A full description of the campaign, including the filtering of the annotations, is detailed in the supplementary material.

Faithfulness	**Gold**	**Ours**	**MeanSum**
Correct	67	47	43
Incorrect	3	7	16
%Correct	95.71	87.04	72.88

Table 4: Results of the human evaluation focused on faithfulness of generated reviews.

The results in Table 4 show that 87.0% of the generated summaries of our system are considered factually correct (compare with 95.7% for the gold summaries), as opposed to 72.9% of MeanSum.

Ablation We analyzed the impact of our proposed variations of the basic self-supervised setting in Table 3. Removing control codes degrades significantly sentiment and category classification of the produced summary F_1. It also impacts greatly the ROUGE score. Changing the decoder-encoder attention from parallel to mean (Sect. 5) also degrades ROUGE. The difference of this attention change without control codes is smaller but – surprisingly – in the different direction.

Control Codes The previous ablation study shows the importance of the control codes in the quality of the final summaries. In order to see how rigidly the model follows those control codes we devise the following experiment to see if the tokens used as control codes are forced to appear in the output text, independent of the input text.

For this, we sample 500 reviews (for 279 venues from the Yelp validation set). For each input example, we randomly sample 8 inferred control tokens (see Sect 4) from the tokens occurring in the review, referring to these as *correct control tokens*. We run the decoder using these control tokens as prompt and count the proportion of them that also occurs in the generated summary. For comparison, we repeat the same experiment but sampling instead 8 control tokens that do *not* occur in the input text, referring to these as *incorrect*.

To minimize the possibility of conditioning on control tokens that might show up naturally in the generated text, for both settings, we repeat the process 5 times per input example (resulting in 2500 with *correct control tokens* as prefix and 2500 using *incorrect*). We report in Fig. 4 the proportion of fed control codes that are generated by the model in both cases. We observe that the model tends to comply with the correct control tokens that occur in the input documents (eg: 89% of the summaries contain more than 50% of the control tokens), but tends to ignore the control tokens when they do not occur in the input. We illustrate this behaviour with a set of examples generated from the same input but different control tokens in the supplementary material.

8 Conclusion

The promise of unsupervised multi-document abstractive summarization is been hampered by the complexity of those models and the problem of hal-

lucinations. Our proposed model has the advantage of being very simple to train compared to previous proposals. In addition, the combined use of control-codes to steer the generation and of multi-input transformers results in summaries that are better (as measured by automatic measures), and produce more faithful summaries (as measured by human evaluation).

While the generated reviews are more factual than those generated by other models, we want to stress that inaccuracies can still appear. Generated summaries are often conjugated in first person, which could lend to believe that an actual human wrote those. We recommend strongly that any use of such algorithms to be accompanied by a clear disclaimer on its true nature.

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, ACL 2020, Online, July 5-10, 2020*, pages 1934–1945. Association for Computational Linguistics.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 936–945.

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.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Federico Barrios, Federico López, Luis Argerich, and Rosita Wachenchauzer. 2015. Variations of the similarity function of textrank for automated summarization. In *Argentine Symposium on Artificial Intelligence (ASAI 2015)-JAIIO 44 (Rosario, 2015)*.

D.M. Blei, A.Y. Ng, and M.I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

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.

Yun Chen, Victor O. K. Li, Kyunghyun Cho, and Samuel R. Bowman. 2018. A stable and effective learning strategy for trainable greedy decoding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 380–390.

Eric Chu and Peter J. Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 1223–1232.

Maximin Coavoux, Hady Elsahar, and Matthias Gallé. 2019. Unsupervised aspect-based multi-document abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. 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 1129–1141, Minneapolis, Minnesota. Association for Computational Linguistics.

Hady ElSahar and Samhaa R. El-Beltagy. 2015. Building large arabic multi-domain resources for sentiment analysis. In *Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part II*, pages 23–34.

Hady ElSahar, Christophe Gravier, and Frédérique Laforest. 2018. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 218–228.

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.

Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220.

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.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

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.

Demian Gholipour Ghalandari. 2017. Revisiting the centroid-based method: A strong baseline for multi-document summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 85–90, Copenhagen, Denmark. Association for Computational Linguistics.

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

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1535–1546.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 225–233.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq R. Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. Gedi: Generative discriminator guided sequence generation. *CoRR*, abs/2009.06367.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1192–1202.

Jindrich Libovický, Jindrich Helcl, and David Marecek. 2018. Input combination strategies for multi-source transformer decoder. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 253–260.

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.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881.

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.

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 11–19.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2014. Labr: A large scale arabic sentiment analysis benchmark. *arXiv preprint arXiv:1411.6718*.

Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2018. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1191–1204, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Andrew Y Ng. 2004. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011a. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011b. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Maxime Peyrard. 2019. A Simple Theoretical Model of Importance for Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1059–1073, Florence, Italy. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1314–1324.

Dragomir R. Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919 – 938.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of*

the first instructional conference on machine learning, volume 242, pages 133–142. Piscataway, NJ.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.

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.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1702–1723.

Ori Shapira and Ran Levy. 2020. Massive multi-document summarization of product reviews with weak supervision. *arXiv preprint arXiv:2007.11348*.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

V Vapnik and A Lerner. 1963. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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.

Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. BottleSum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3750–3759, Hong Kong, China. Association for Computational Linguistics.

Lijun Wu, Yingce Xia, Fei Tian, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. Adversarial neural machine translation. In *Proceedings of The 10th Asian Conference on Machine Learning, ACML 2018, Beijing, China, November 14-16, 2018*, pages 534–549.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 7298–7309.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic VQA: disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 1039–1050.

Li Yuan, Francis EH Tay, Ping Li, Li Zhou, and Jiashi Feng. 2019. Cycle-sum: cycle-consistent adversarial lstm networks for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9143–9150.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pages 9051–9062.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

A Generated Examples

Figures 6, and 7 include a set of samples generated from our model and baselines, full generations our model can be downloaded from https://www.dropbox.com/s/w6eqviy5fnda11f/hypos_and_refs.zip?dl=0

B Control Codes Analysis

Figure 5 illustrates the behaviour of the model on the same input when it is fed either 'correct' or 'incorrect' control codes. The correct control codes occur in the output, but this is not the case for incorrect control codes: the model does not simply learn to copy control codes, it ignores them when they are inconsistent with the input reviews.

Correct Control Tokens: ***eat***, ***lentil***, ***eggplant***, ***new***, **remember,** ***flavourful***, ***friendly***

Seriously best we've had in Toronto. We were looking for a *new* place to *eat* and stumbled upon this place. The atmosphere is very authentic, the food is authentic, and the service is very *friendly* and attentive. We started with the *lentil* soup, which was very *flavourful* and full of flavor. For dinner, I had the lamb shank and my husband had the *eggplant* dish. Both were very good. We also had the baklava for dessert and it was amazing. We can't wait to come back and try more of the menu

---

Correct Control Tokens: ***yogurt***, ***remember***, ***service***, ***shawarma***, ***dishes***, ***eat***, ***restaurant***, ***flavourful***

Seriously best I've had in Toronto. We were looking for a place to *eat* and this was the first *restaurant* we went to. The *service* was great and the *dishes* were authentic, *flavourful* and authentic. We had the *shawarma* sampler and the *yogurt* sampler. Both dishes were amazing. We also had the baklava, which was also very good. I can't *remember* the name of the dish, but it was very authentic

---

Incorrect Control Tokens: open, tortellini, filter, details, rabbit, signature, craving, double

Seriously best I've had in Toronto. We were looking for a new place to try and stumbled upon Syrianian food. We came across this place on yelp and decided to give it a try. The food, the service, the atmosphere, the quality of the food, and the service were all amazing. We ordered the sampler platter for two and it was enough for two people to share. We also ordered the Brabianianian appetizer, which was one of the best we've ever had. For our main course, we ordered the Crabian Frabian Bianianca and the Bianca. Both were so authentic, and full of flavour. For dessert, we had the Turkish coffee which was also very good. The only thing I didn't like was the bread they give you .

Incorrect Control Tokens: beer, balls, skip, mains, croissant, replacement, wish

Seriously best I've had in Toronto. We came here with a friend and we were looking for a place to eat. We found Syrianian food, which was very authentic. We were very impressed with the quality of the food, the service, and the atmosphere. We had the sampler platter, which came with two mains and two mains for dinner. We also had the bread pudding for dessert and it was to die for. I'm not a huge fan of sweets, but this was one of the best we've ever had. I wish we lived in Toronto so we could come here all the time. We'll be back to try more of the menu .

Figure 5: Summaries generated **from the same input** when different 'correct' and 'incorrect' control tokens are fed as prefixes at inference time. Control tokens that occur in the summary are highlighted (green/italics for the first rows, red/underlined for the other two).

C Inferred Control Tokens

Fig. 8 shows examples of the top inferred tokens for some categories in the Yelp dataset, those tokens have been inferred using our proposed method in this work.

D Human Evaluation Campaign

We used Amazon Mechanical Turk to ask 3 "workers" to assess if 282 summaries produced by 3 systems (94 from each: ours, gold from human experts and MeanSum) aligned correctly with sets of 8 reviews. Workers had to read the reviews, the summary and answer the question: "does the summary contain correct information given in the original reviews?" Instructions specified to "assess the faithfulness of the summary with respect to the set of reviews," specifically to "verify that the summary did not contain factually incorrect or self-contradicting statements that could not be inferred from what was provided in the original reviews." Using Mechanical Turk qualification criteria, we asked for the workers: (1) to be located in the United States, Canada or United Kingdom; (2) to have a HIT approval rate higher than 98; (3) to have more than 1000 HITs approved.

Note that in an initial pilot, we asked evaluators to pick the best and worst summaries for fluency, coherency and alignment, as well as overall. We decided to simplify the task because it turned out to be a quite difficult one as workers struggled on many summaries to decide of the best and worst. We decided to focus the human evaluation on the aspect that is currently very difficult to automate, faithfulness to the original text. We see this evaluation as complementary to the automatic evaluation, focusing on different aspects.

We did an internal run to estimate the time needed per individual assignment – each Human Intelligence Task, or HIT, an annotation in our case, was assigned to 3 workers. We followed it by a short pilot to validate the average 2 minutes we had estimated. This is important to establish the rate to pay: 2 minutes translate into 30 potential assignments per hour, we picked $0.50 to target an average $15 hourly wage. Beyond the timing, the pilot was also used as a dry run for the full campaign. The average time to answer and the theoretical hourly wage are provided in Table 6

By using shuffled gold summaries, hence written for another set of reviews, we included 21 badly aligned "negatives." Workers who answered *yes* for these obvious *no* were filtered out as "dubious" from the results: all their answers were dis-

Dataset	Reviews		Businesses / Movies
	Train	Valid	
Yelp — Ours	349,839	48,677	22,522
Yelp — Ours + Control	404,811	47,938	22,522
Rotten Tomatoes — Ours	167,168	18,731	3,732

Table 5: Sizes of Training and validation splits of different datasets.

carded. After filtering out the "negatives" HITs and the ones from "dubious" answers, we were left with 446 annotations, from the 782 we received. We further discarded all annotations made in less than a minute to keep 377 realistic answers —one minute may seem harsh but we estimated it was the minimum time needed to first read the reviews, then the summary, and to assess the latter, given the question: proceeding backward by first reading the summary would still require the worker to read all summaries, to make sure a factual error according to one review is not extracted from another one.

Finally we looked for full agreement at the HIT level and kept only the ones with either 0 *yes* or 0 *no*, with varying numbers, from 1 to 3, of the alternatives after the filtering of the "dubious" and "unrealistic" answers. Not surprisingly, as we focused on alignment, Gold summaries scored best but ours scored nicely, with a very low number of misaligned summaries.

Assessing the alignment of summaries to a set of reviews is not an easy task. We decided to discard all answers from the "dubious" workers who erred on our "negatives" summaries to be on the safe side. Mechanical Turk reports the time taken for an assignment, their averages is an interesting metric to look at, especially the way it evolves along our filterings — we translated it to the associated theoretical hourly wages, alas all under the $15 we initially targeted.

We also looked at the results with no full agreement: instead of doing it per HIT, or summary, it had to be done at the lower level of the evaluation. For the 276 evaluations with full agreement, the numbers are: Gold 108/4 (96.43% correct), Ours 69/7 (90.79%), MeanSum 63/25 (71.59%).
When including the disagreements (377 evaluations), they are: Gold 116/13 (89.92%), Ours 89/27 (76.72%), MeanSum 85/47 (64.39%).
The numbers are similar, however given the difficulty of the assessment for the workers, we decided to focus on the summaries they agreed on.

Inputs:
1. Best Philly Ever!!! Thank You Sam!!! Sometimes it is the little things in life that can Make You Happy- All it took was a Perfect Cheese Steak to Cheer Me Up, not to mention seeing a Friend Again - Thanks again Sam,, It wouldn't be the same without You
2. Wow after all the hype about what a great place I was really disappointed. If this is a franchised operation than the quality control is really lacking. Our first visit to Capriotti's and with so many other quality places I doubt if they will get us as repeat customers. Well, here it is. We ordered the Bobbie and the Capastrami shared it. Both had cold bread in fact we got the impression that both sandwiches had been pre made and put in a refrigerator because the insides were also cold. No taste at all in either. For a company that supposedly cooks overnight you would think the turkey ingredients would look like turkey but apparently they shred it into little tiny bits. Will not return
3. This place is always good, I think the owner actually made my sandwich last time I was there , owner or manager, anyway it was superb! quite flavorful, even the next day it tasted just as good. Grab a Capistrami you can't go wrong, until next time Cappie's , be well.
4. one New Year's resolution is to write more Yelp reviews, so here goes... In Vegas for NYE and gave this place a shot per other Yelp reviews. I had the Capistrami and the girlfriend had the Cheese Steak, which I had few bites of. Both were absolutely delicious in an awesome-deli-sandwich-sort-of-way. The shop is no-frills with only some bar seating, but the sandwiches are really reasonably priced. So if all you're after is a fantastic deli sandwich, definitely go.
5. number 1 in Vegas for a reason. Everyone has their favs.... the capistrami, the cheese steak with mush... Mine is definitely the Bobby. In case you haven't viewed their menu yet, the bobby is thanksgiving leftovers in a huge sandwich... yeah, exactly.
6. Worst service I have seen at a capriotti's. Ordered a sandwich, watched the guy write it down and 25 minutes later the same person asked what I wanted when I reminded him of my sandwich, he only said he can't remember where the order went. I watched 4 people come in after me order, one person the same sandwich just a different size then me get their food, pay and leave. I will not be coming back to the location ever again. Looks like I will be going to firehouse for now on!
7. Stopped in for a sandwich on the way to the park. Next day I notice the charge has had a $2 tip added to it that I did not authorize. (I left a cash tip in the beer money jar) I called Corporate and got nowhere because this is a franchise store. At that point I gave up because as much as I like their sandwiches I am never going back.
8. Don't bother calling in an order. If they tell you a time it will be off by at least thirty minutes. Terrible service. Great food tho.

Summary OURS:
This was my *first visit* to *Capriotti's* and I really enjoyed it . I had the *Capastrami* and my husband had *the Bobbie* . We both enjoyed our *sandwiches* as well . The *quality* of the *ingredients*, however, was not what we expected . We also enjoyed the *cheese steak* as well as the *turkey*, which was not bad at all . This place is a bit on the expensive side for what you get, but you get what you pay for . The *seating* is limited, so it's a good place to visit if you're in a hurry.

Summary MeanSum:
Drove by here for the *first time*. I just went to the *deli* with a friend and it's a quick fix that is just about as good as it gets. But it's not an actual *sandwich*, but it's not as good as I remembered it, but they were great!! *Sandwich* was also very good, just a hint of cinnamon. I will be back for the other locations.

Summary TextRank (Extractive):
Will not return This place is always good, I think the owner actually made my sandwich last time I was there , owner or manager, anyway it was superb! Ordered a sandwich, watched the guy write it down and 25 minutes later the same person asked what I wanted when I reminded him of my sandwich, he only said he can't remember where the order went. I watched 4 people come in after me order, one person the same sandwich just a different size then me get their food, pay and leave. At that point I gave up because as much as I like their sandwiches I am never going back.

Figure 6: Examples of output summaries for different models.

Set	Unfiltered	Negatives discarded	Dubious discarded	less than 1min discarded	Agreement
Average time to Answer	2min16s	2min17	2min9	2min26	2min26
Theoretical hourly wage	13.22	13.16	13.96	12.36	12.35

Table 6: Average time to Answer and the theoritical hourly wage of turkers (in USD) for the crowdsourcing experiments of human evaluation.

Inputs:
1. Great service and a super clean nice location here. Considering this is in a busy airport, I was impressed. The pricing here, which is about double to triple regular prices, was what knocks off a star for me.
2. If you're by the D gates at Sky Harbor, this is your coffee stop. Much better than the Starsucks at the high C gates.
3. Spotted in due to flight delay. The big comfy brown lawyer seats is what attracted me in. I figured I could get some work done. I ordered a non fat vanilla iced tea. It was pretty good. I noticed the prices were a lil bit more expensive.
4. $9 for a sandwich. I guess you can charge whatever you want when the airport doesn't have any other options in concourse C
5. The line is quick, the people are friendly and the drinks are tasty. Also for skyharbor employees, they actually give an airport discount, unlike Starbucks.
6. Try gingerbread latte yum. This is the best looking, most comfortable airport coffee shop I've ever been in !!! Big comfy chairs with little tables. A big water container with cups in the restaurant away from the congestion of the order counter. Friendly , happy workers equals happy customers. I know you can't please everyone, but at 5am a room full of happy airline travelers is a hard thing to come by. If your in need of coffee in Phoenix Sky Harbor , terminal 3 be sure to stop by and take a load off !!
7. The lid fell off my cup, burned my hand, and spilled half of my coffee. Employees never asked if I was okay, or offered to replace my coffee. Will not be back to this location.
8. Delicious cup of coffee. Very impressed Mr. Peets. Will be returning whenever I can.

OURS:
Try the *gingerbread* cups. The coffee and ginger cups are *delicious* and the *chairs are comfortable* . I've been working in the *terminal* for a long time . This is a must stop if you're in the *airport* . The staff is *friendly* .

MeanSum:
5.50 for a 2.5" breakfast sandwich. I'm a big fan of the concept but this place is way better than *Starbucks*. The staff is *friendly*, and fast. I'm not a big fan of sweets but I'd be happy to come back.

TextRank (Extractive):
Great service and a super clean nice location here. Considering this is in a busy airport, I was impressed. The pricing here, which is about double to triple regular prices, was what knocks off a star for me. I noticed the prices were a lil bit more expensive. I guess you can charge whatever you want when the airport doesn't have any other options in concourse C The line is quick, the people are friendly and the drinks are tasty. This is the best looking, most comfortable airport coffee shop I've ever been in !!! Will not be back to this location.

Figure 7: Examples of output summaries for different models.

Delis: deli, sandwiches, sandwich, bagels, skinnyfats, subs, bagel, sub, chompie, smoked meat
Nail Salons: nails, pedicure, nail, pedicures, pedi, salon, manicure, pedis, colors, salons
Sushi Bars: sushi, hibachi, kona, rolls, roll, japanese, ayce, sake, benihana, poke
Florists: flowers, trader, arrangement, florist, wedding, bouquet, tj, arrangements, aj, grocery
Beauty & Spas: walgreens, tattoo, sephora, ti, vdara, tattoos, haircut, barbers, barber
Party & Event Planning: herb box, wedding, kids, fun, party, event, golf, painting, rainforest, blast
Trainers: gym, workout, fitness, equipment, membership, trainers, training, trainer, instructors, machines
Cafes: cafe, first watch, bouchon, salsa bar, café, coffee, breakfast, gallo, crepes, latte
Mags: books, store, book, games, bookstore, selection, records, comics, vinyl, game
Ice Cream & Frozen Yogurt: gelato, ice, sonic, yogurt, custard, culver, flavors, freddy, froyo, icecream
Burgers: burgers, burger, mcdonald, ihop, applebee, red robin, mcdonalds, wellington, hamburgers, castle
Furniture Stores: furniture, ikea, mattress, store, sales, delivery, bought, couch, purchase, bed
Sporting Goods: bike, bikes, shoes, gear, gun, store, range, golf, shop, equipment
Bakeries: bakery, pastries, wildflower, cupcakes, panera, cake, pastry, cookies, cinnamon rolls, cakes
Thai: thai, curry, pad, asian, khao, curries, food, papaya, satay, tom
Gyms: gym, workout, fitness, membership, equipment, trainer, trainers, work out, coaches, class
Cosmetics & Beauty Supply: walgreens, pharmacy, products, haircut, store, sephora, hair, makeup, lashes, kohl
Auto Repair: car, vehicle, dealership, cars, auto, mechanic, vehicles, oil, windshield, tire
Department Stores: walmart, target, costco, store, department, shopping, mall, section, ross, sears
Local Services: post office, thrift, laundromat, daycare, guitar, cleaners, pest, activities, storage, laundry
Hair Extensions: hair, salon, stylist, color, haircut, extensions, appointment, she, lashes, blow
Hair Removal: eyebrows, nails, pedi, pedicure, nail, appointment, brows, wax, salon, waxing
Laundry Services: cleaners, clothes, laundry, cleaning, dry, laundromat, dress, pants, machines, shirts
Doctors: dr, doctor, doctors, medical, hospital, office, patients, appointment, nurse, clinic
Movers: move, moving, movers, company, truck, storage, guys, furniture, moved, haul
Printing Services: printing, print, ups, package, business, fedex, shipping, customer, printed, store
Makeup Artists: makeup, hair, salon, make up, lashes, stylist, blow, appointment, eyebrows, brows
Plumbing: plumbing, company, plumber, water, call, called, work, house, job, leak
Real Estate Services: property, estate, westgate, home, company, process, house, realtor, rent, work with

Figure 8: Examples of Inferred control tokens for each category of venues for the Yelp dataset.