Abstractive Multi-Document Summarization Through Self-Supervision and Control

Anonymous Authors[1]

Abstract

We address the problem of unsupervised abstractive summarization of collections of user generated reviews with a novel self-supervised setting. In this setting, an individual document is considered as a target summary for a set of similar documents making training simpler than previous approaches by relying only on standard log-likelihood loss with teacher forcing. We modify the Transformer architecture to allow for multiple reviews as input, and 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 datasets against graph-based and recent neural abstractive unsupervised models show that our proposed method generates summaries with a superior quality and relevance, which is confirmed in our human evaluation. We also provide an ablation study, which shows the importance of the control setup in controlling hallucinations and achieve high sentiment and topic alignment of the summaries with the input reviews.

1. Introduction

Recent progress in unsupervised methods have 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 with a reconstruction loss. Machine translation is the most successful of those applications, but 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 in 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 & Tarau, 2004). Only recently have there been proposals for unsupervised abstractive summarization, using auto-encoders (Chu & Liu, 2019; Bražinskas et al., 2019). However, these setups are quite complex, requiring a combination of loss functions (Chu & Liu, 2019) or hierarchical latent variables (Bražinskas et al., 2019) 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 a multiple of 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 utmost 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., 2017; Keskar et al., 2019). Control tokens are discrete variables that are used to condition the generation. Different from previous work, our goal by using those 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 based on a recently proposed 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

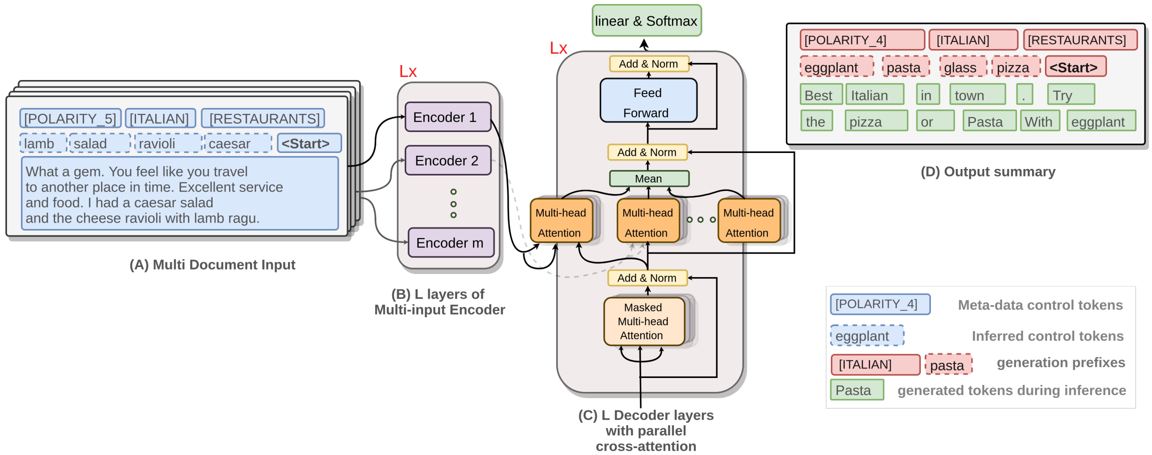

Figure 1. Description of our proposed model: (A) is the set of input reviews, augmented with control tokens (from meta-data in uppercase, inferred in lowercase). (B) is the encoder, which is run separately on each input review. The standard Transformer decoder is modified in (C) to allow for Parallel 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.

time applies parallel attention to each encoded input (Sect. 5).

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

2. Related Work

Unsupervised Opinion Summarization Unsupervised Multi-Document summarization methods encompass both *extractive* and *abstractive* approaches. Extractive summarization consists in selecting a few sentences from the input documents to form the output summary. Radev et al. (2004) proposed to rank sentences according to their relevance to the whole input, representing sentences as tfidf bags of words and the input as the centroid vector of its sentences. Recent refinements of this approach include using distributed word representations (Rossiello et al., 2017) or ranking whole summaries instead of individual sentences (Gholipour Ghalandari, 2017). Graph-based methods, such as LexRank (Erkan & Radev, 2004) or TextRank (Mihalcea & Tarau, 2004; Zheng & Lapata, 2019), work by constructing a graph whose nodes are the sentences from the input documents and whose edges indicate a high word overlap between two sentences. Then they use the PageRank algorithm to extract the sentences with the highest centrality. In contrast to these methods, we focus on abstractive summarization methods.

Abstractive methods for summarization are in principle able to generate new words and sentences that do not occur in the input documents and therefore enhance produce more fluent text. Non-neural abstractive methods (Ganesan et al., 2010; Nayeem et al., 2018) are also graph-based, but construct graphs whose nodes are word types and edges indicate the immediate precedence relationship between two instantiations of the word type in a sentence. The summary is extracted by finding salient paths in the graph.

Recently, a few approaches for unsupervised neural abstractive summarization have been proposed. Chu & 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' vectors 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 leading it to rely on only language modeling for generating the output summary and ignore the input signal. To deal with this limitation, Coavoux et al. (2019) proposed to add a clustering step to identify similar reviews and to generate one sentence per such found cluster: the averaging only targets similar reviews. Contemporaneous to this work Bražinskas et al. (2019) proposed to solve the problem of unsupervised summarization of reviews through an auto-encoder with latent variables. Their proposed way of solving the problem of hallucinating content from other categories is to use a latent variable per product, and let the decoder access all the reviews of a product. Compared to it, we argue that our self-supervised setting is simpler as it relies on training with standard cross-entropy. In addition the use of Transformer (as opposed to GRU in their case) makes it possible to apply separate attentions to each input.

West et al. (2019) introduced a self-supervised system for sentence compression: they design an unsupervised extractive system and use it to generate data to train a supervised neural sentence compressor. However, their two-level system works at the level of single sentences whereas our end-to-end approach summarizes sets of reviews with multiple sentences.

Controlled Generation We rely on controlled natural language generation to steer the generation away from hallucinations. Controllable text generation has been previously investigated to apply global constraints on text generation. Previous work proposed fine-tuning NLG models to provide control. To allow back-propagation through the discrete sampling process of text generation several proposals have used Policy gradient methods, most notably REINFORCE (Williams, 1992) for applications such as machine translation (Ranzato et al., 2016; Wu et al., 2018), image-to-text generation (Liu et al., 2017), dialogue generation (Li et al., 2016b) and visual question answering (Yi et al., 2018). Other work has relied on continuous approximation methods, most notably the Gumbel-Softmax (Jang et al., 2017) such as in Chu & Liu (2019); Yang et al. (2018).
Other methods of control applied control only at inference time using weighted decoding (Holtzman et al., 2018), which was shown to be challenging and to often lead to sacrificing fluency and coherence (See et al., 2019), or constrained beam search (Anderson et al., 2017; Hokamp & Liu, 2017; Post & Vilar, 2018) which is slower, requires in practice very large beam sizes and does not allow soft constraints, or finally by updating the decoder hidden states (Chen et al., 2018; Dathathri et al., 2019) which requires an extra training step.
The first introduction of control codes to neural generation models has been an early form of copy mechanism to overcome the rare word problem (Luong et al., 2015; ElSahar et al., 2018) and recently has shown a wide adoption - due to its simplicity and effectiveness - to steer large scale language models toward desired traits such as general aspects (Keskar et al., 2019) or structured fields (Zellers et al., 2019).
Previous work for controlling large-scale language models has relied on a predefined set of bag of control tokens collected either manually (Keskar et al., 2019) or from dictionaries (Dathathri et al., 2019) this can yield to low domain coverage. Regularized classification models have intrinsic feature selection capabilities (Ng, 2004) this has been exploited before for lexicon generation from sentiment classifiers (Nabil et al., 2014; ElSahar & El-Beltagy, 2015), such approaches have proven to generate more relevant lexicons than traditional topic models such as LDA (Blei et al., 2003).
In this work, to automatically generate bag of control tokens we follow the same approach which does only rely on the category meta-data provided with the reviews.

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 explore 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 be fixed (k), and fixing s_i we look for a set of k *good* reviews D_i for which s_i serves as good summary. Fixing k also simplifies training and allow comparison with previous work where the number of input reviews is fixed (Chu & Liu, 2019; Bražinskas et al., 2019). Both s_i and all members of D_i are reviews of the same entity.

To solve the second problem of defining what a good set of reviews for a given summary s_i is, we use the recently proposed theoretical model of importance in summarization (Peyrard, 2019) which defines the importance of a summary based on three aspects: minimum redundancy, maximum relevance with the input document and maximum informativeness. Redundancy and informativeness are not dependent on D_i, therefore we focus solely on finding a *relevant* set of review D_i for the summary s_i (instead of the opposite, as is the original motivation of Peyrard (2019)).

We define *rel*(s_i) as the set D_i of size k that maximizes the similarity between s_i and each member of D_i:

$$\begin{aligned} \textit{rel}(d_i) &= \{d_{i_1}, d_{i_2}, ..., d_{i_k}\}, \\ &= \underset{D_i \subset \mathcal{D} \setminus \{d_i\}, |D_i|=k}{\arg\max} \sum_{d_j \in D_i} \mathrm{sim}(d_i, d_j). \end{aligned}$$

The data-points $(d_i, \textit{rel}(d_i))$ are then sorted according to the value of the relevance $(\sum_{d_j \in \mathrm{rel}(d_i)} \mathrm{sim}(d_i, d_j))$. According to the desired size of the target dataset, the top-T of the pairs are then retained for training. Limiting T inherently increases informativeness as 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."). In addition to simplicity, this method has the advantage of allowing a fast implementation using state-of-the-art 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 for instance if the synthetically created training data contains a variety of contradictory signals, or because certain types of review are overly present (e.g. "great movie"). The model might default to those very frequent patterns if during decoding time it finds itself in a unfrequent state.

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. We use two types of metadata that represent the review **polarity**, a numerical value denoting the average sentiment score of the input reviews; and **categorical tokens** representing the type of the entity of the review (e.g. Deli, Beauty&Spa, Furniture Stores). In the case of the unavailability of meta-data labels for all reviews (as in Rotten-Tomatoes dataset) we infer control tokens with the same process, but using categories predicted by a trained classifiers on a subset of labeled example.

2) Inferred control tokens. We follow recent work (Keskar et al., 2019; Dathathri et al., 2019) 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 & 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

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

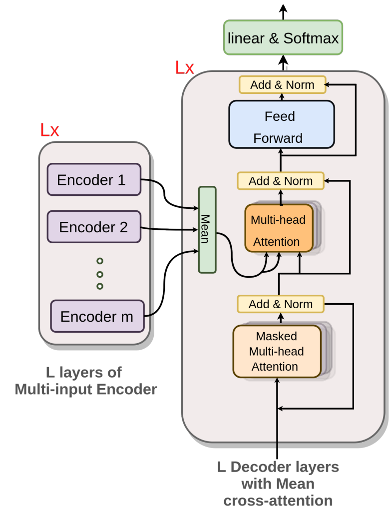

Figure 2. Figure showing our adaptation of the Transformer cross-attention to allow *Mean* combination of multi-sources.

number of features (in our experiments were all unigrams and bigrams present in the corpus). We used a squared hinge loss with $L1$ regularization over θ_i – 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 re-normalize 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 $\theta_{\ell_i^{(p)}}$. 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, each review is enriched with its tailored control codes. In particular, the reviews acting as summary also contain them, and by construction those are n-grams present in the text. At inference time – when the target side and hence 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 before the start of generation. 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 & Liu, 2019) built multi-source input representations through a simple mean over the last hidden states of the encoder. This proposal has limitations coming from representing the full set of reviews 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 ($K_i \in K_{i:m}$) and values ($V_i \in V_{i:m}$) are generated and then the yielded context is averaged and followed by a residual connection from the previous decoder layer. This corresponds to 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 input combination strategy, which is less computationally demanding. It does not apply the cross-attention with each encoder separately and 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.

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

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 model 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 & 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 longest model took 50 hours to train. For inference, we use a beam size of 35. We discard hypotheses that contains 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 Chu & 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 & Liu, 2019) and Rotten Tomatoes (Wang & Ling, 2016). The Yelp dataset contains reviews of businesses (approximately one million reviews for around 40k venues). As described in Section 3, for each venue, we select the best candidates to use as proxy summaries: either the top-p (with $p = 15\%$) or the top-T (with $T = 100$) reviews, whichever is smaller. For each proxy summary thus selected, we then take its 8 most similar reviews (cosine similarity) to form its input. We obtain around 340k training examples, representing 22.5k venues.

The Rotten Tomatoes dataset was constructed by (Wang & 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. Details about dataset sizes and splits are in the Appendix A

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., 2020), 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 indicate the diversity of different generations. Finally, as done by Chu & Liu (2019), we use a classifier to check whether the sentiment of the summary is consistent with the sentiment of input reviews (Sentiment Acc. in Table 1).[4] We extend this method to

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

[3] For Yelp we use the Meansum (Chu & Liu, 2019) implementation to keep results comparable while for RottenTomatoes we use *py-rouge* package `pypi.python.org/pypi/pyrouge/0.1.3`

[4] We use a 3-class classification: negative (1 or 2 star), neutral (3), positive (4 and 5)). As a result, the numbers are not comparable

	Model	ROUGE-1	ROUGE-2	ROUGE-L	F_{BERT}	Sentiment Acc.	$F_{category}$
YELP	Textrank (Mihalcea & 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., 2019)	29.5	5.3	18.1	–	–	–
	Meansum (Chu & 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**

Table 1. Automatic evaluations results against gold summaries of Yelp and Rotten Tomatoes datasets. The F_{BERT} score is described by Zhang et al. (2020). "Ours" denotes our proposed system with parallel input combination strategy and control codes.

	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.

check whether the correct product category can also be inferred from the summary ($\mathbf{F}_{category}$).

Baselines and Other Systems We compare our system with three unsupervised baselines. TextRank (Mihalcea & Tarau, 2004) and LexRank (Radev et al., 2004) are extractive systems based on the PageRank algorithm. Opinosis (Ganesan et al., 2010) is an abstractive graph-based system. We use openly available Python implementations for TextRank[5] (Barrios et al., 2016) and LexRank.[6] We use the default parameters of the implementations. For Opinosis, we use the official with default hyperparameters.[7]

We also compare our systems with more recent neural unsupervised summarization systems. For the Yelp dataset, we rerun the released pretrained version of Meansum[8] (Chu & Liu, 2019).

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, inlcuding the recent neural unsupervised models of Meansum and H-VAE. Note that because of the concurrent nature of the Bražinskas et al. (2019) paper, the H-VAE model is not available and we report the numbers from their paper.[9] For Meansum we re-run their provided checkpoint and run evaluation through the same pipeline. The BERTScore (Zhang et al., 2020) 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, meaning, our model is capable of generating more rich and less repetitive summaries. On the level of all generations our model generates text with more diversiy than Meansum. In gen-

with those reported Chu & Liu (2019).

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

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

[7]Except for the redundancy parameter which was set to one, since the default led to many empty outputs.

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

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

Model	*Quality*						*Speed*
	ROUGE-1	**ROUGE-2**	**ROUGE-L**	$\mathbf{F}_{\text{BERT}}$	**Sentiment Acc.**	$\mathbf{F}_{\text{category}}$	**Train.** (wps)
$\text{Ours}_{\text{Parallel}}$	**32.8**	**8.7**	**18.8**	86.8	**83.9**	55.2	3785
$\text{Ours}_{\text{Mean}}$	29.4	5.3	17.2	**87.6**	83.4	**56.2**	8075
$\text{Ours}_{\text{Parallel}}$ − cntrl.	25.3	3.7	15.5	85.2	76.9	43.9	7609
$\text{Ours}_{\text{Mean}}$ − 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

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 denotes substrings with exact match with the input, red 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.

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 4. Examples of outputs 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 for the first rows, red for the other two).

eral 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.

Surprisingly, Meansum performs very close to the graph-based models, while our proposed model seems to produce more novel content. 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", this is possibly attributed to the cross-attention mechanism in our proposed model. More examples are provided in the supplementary material Appendix D.

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 crowd-sourcing 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

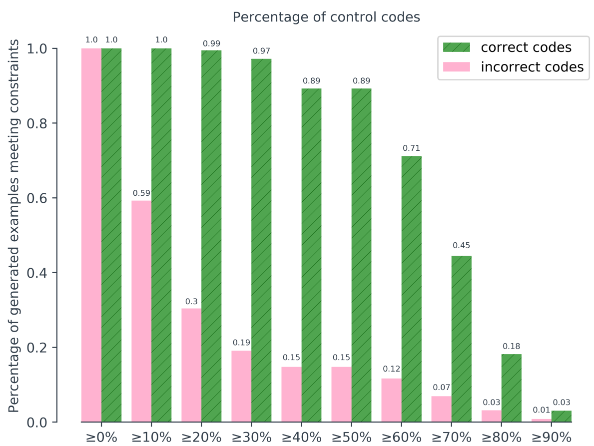

Figure 5. Proportion of control codes fed as prompts that occur in the generated summary, for the setting of *correct* and *incorrect* control tokens.

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 Appendix F.

Faithfulness	**Gold**	**Ours**	**Meansum**
Correct	67	50	47
Incorrect	3	4	12
%Correct	95.71	92.59	79.66

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

The results in Table 4 show that 92.6% of the generated summaries of our system are considered factually correct (compare with 95.7% for the gold summaries), as opposed to 79.7% 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 – as expected – sentiment and category classification of the produced summary F_1. It also impacts greatly on the ROUGE score. Changing the decoder-encoder attention from parallel to mean (see 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 set of input reviews (for 279 venues from the Yelp validation set). For each input example, we randomly sample 8 control tokens (inferred control codes, see Sect 4) among those present in the example. We refer 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. We refer to these as **"incorrect control tokens"**.

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 3000 with *correct control tokens* as prefix and 3000 using *incorrect*). We report in Figure 5 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. Fig. 4 shows a set of generated examples for the same input when the model is conditioned on different control tokens.

8. Conclusion

Neural methods have shown great promises for abstractive summarization, overcoming the lack of fluency of extractive models. However, those models are often complex and more importantly tend to generate incorrect statements; characteristics which are exacerbated in the unsupervised setting. Our proposed models aim to overcome those problems by proposing a simple training mechanism relying on a self-supervised formulation. In addition with our use of multi-input transformers and control codes we show that the resulting summaries are better (as measured by ROUGE and other automatic measures), as well as producing more faithful summaries (as measured by human evaluation). The use of control codes make it easy to extend for other multi-document summarization use-cases.

While the generated reviews are more factual than those generated by other models, we want to stress that inaccuracies can still appear and that special care should be taken if such methods are to be deployed. In particular, the models learn the conjugations from the input, which is mostly in first persons. Such summaries might be misleading as it 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

Anderson, P., Fernando, B., Johnson, M., and Gould, S. 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*, pp. 936–945, 2017. doi: 10.18653/v1/d17-1098. URL https://doi.org/10.18653/v1/d17-1098.

Artetxe, M., Labaka, G., Agirre, E., and Cho, K. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=Sy2ogebAW.

Barrios, F., López, F., Argerich, L., and Wachenchauzer, R. Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606, 2016. URL http://arxiv.org/abs/1602.03606.

Blei, D., Ng, A., and Jordan, M. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. URL http://www.cs.princeton.edu/~blei/lda-c/.

Bražinskas, A., Lapata, M., and Titov, I. Unsupervised Multi-Document Opinion Summarization as Copycat-Review Generation. *arXiv:1911.02247 [cs, stat]*, November 2019. URL http://arxiv.org/abs/1911.02247. arXiv: 1911.02247.

Chen, Y., Li, V. O. K., Cho, K., and Bowman, S. R. 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*, pp. 380–390, 2018. doi: 10.18653/v1/d18-1035. URL https://doi.org/10.18653/v1/d18-1035.

Chu, E. and Liu, P. J. 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*, pp. 1223–1232, 2019. URL http://proceedings.mlr.press/v97/chu19b.html.

Coavoux, M., Elsahar, H., and Gallé, M. Unsupervised aspect-based multi-document abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 42–47, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5405. URL https://www.aclweb.org/anthology/D19-5405.

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., and Liu, R. Plug and play language models: a simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Drozdov, A., Verga, P., Yadav, M., Iyyer, M., and McCallum, A. Unsupervised latent tree induction with deep inside-outside recursive autoencoders. *arXiv preprint arXiv:1904.02142*, 2019.

ElSahar, H. and El-Beltagy, S. R. 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*, pp. 23–34, 2015. doi: 10.1007/978-3-319-18117-2_2. URL https://doi.org/10.1007/978-3-319-18117-2_2.

ElSahar, H., Gravier, C., and Laforest, F. 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)*, pp. 218–228, 2018. doi: 10.18653/v1/n18-1020. URL https://doi.org/10.18653/v1/n18-1020.

Erkan, G. and Radev, D. R. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479, December 2004. ISSN 1076-9757. URL http://dl.acm.org/citation.cfm?id=1622487.1622501.

Falke, T., Ribeiro, L. F., Utama, P. A., Dagan, I., and Gurevych, I. 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*, pp. 2214–2220, 2019.

Fan, A., Grangier, D., and Auli, M. Controllable abstractive summarization. *arXiv preprint arXiv:1711.05217*, 2017.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug): 1871–1874, 2008.

Ganesan, K., Zhai, C., and Han, J. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 340–348, Beijing, China, August 2010. Coling 2010 Organizing Committee. URL https://www.aclweb.org/anthology/C10-1039.

Gholipour Ghalandari, D. Revisiting the centroid-based method: A strong baseline for multi-document summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 85–90, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4511. URL https://www.aclweb.org/anthology/W17-4511.

Hokamp, C. and Liu, Q. 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*, pp. 1535–1546, 2017. doi: 10.18653/v1/P17-1141. URL https://doi.org/10.18653/v1/P17-1141.

Holtzman, A., Buys, J., Forbes, M., Bosselut, A., Golub, D., and Choi, Y. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1638–1649, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1152. URL https://www.aclweb.org/anthology/P18-1152.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Junczys-Dowmunt, M. 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*, pp. 225–233, 2019. doi: 10.18653/v1/w19-5321. URL https://doi.org/10.18653/v1/w19-5321.

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858, 2019. URL http://arxiv.org/abs/1909.05858.

Kryscinski, W., Keskar, N. S., McCann, B., Xiong, C., and Socher, R. 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)*, pp. 540–551, 2019.

Kudo, T. and Richardson, J. 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*, pp. 66–71, 2018. doi: 10.18653/v1/d18-2012. URL https://doi.org/10.18653/v1/d18-2012.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. 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*, 2018. URL https://openreview.net/forum?id=rkYTTf-AZ.

Lewis, P., Denoyer, L., and Riedel, S. Unsupervised question answering by cloze translation. *arXiv preprint arXiv:1906.04980*, 2019.

Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. 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*, pp. 110–119, San Diego, California, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://www.aclweb.org/anthology/N16-1014.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. 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*, pp. 1192–1202, 2016b. doi: 10.18653/v1/d16-1127. URL https://doi.org/10.18653/v1/d16-1127.

Libovický, J., Helcl, J., and Marecek, D. 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*, pp. 253–260, 2018. doi: 10.18653/v1/w18-6326. URL https://doi.org/10.18653/v1/w18-6326.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches*

Out, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Liu, S., Zhu, Z., Ye, N., Guadarrama, S., and Murphy, K. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*, pp. 873–881, 2017.

Luong, T., Sutskever, I., Le, Q. V., Vinyals, O., and Zaremba, W. 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*, pp. 11–19, 2015. doi: 10.3115/v1/p15-1002. URL https://doi.org/10.3115/v1/p15-1002.

Mihalcea, R. and Tarau, P. TextRank: Bringing order into text. In *Proceedings of EMNLP 2004*, pp. 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-3252.

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

Nayeem, M. T., Fuad, T. A., and Chali, Y. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1191–1204, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1102.

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

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011b.

Peyrard, M. A Simple Theoretical Model of Importance for Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1059–1073, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1101. URL https://www.aclweb.org/anthology/P19-1101.

Post, M. and Vilar, D. 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)*, pp. 1314–1324, 2018. doi: 10.18653/v1/n18-1119. URL https://doi.org/10.18653/v1/n18-1119.

Radev, D. R., Jing, H., Styś, M., and Tam, D. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919 – 938, 2004. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2003.10.006. URL http://www.sciencedirect.com/science/article/pii/S0306457303000955.

Ramos, J. et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pp. 133–142. Piscataway, NJ, 2003.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. 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*, 2016. URL http://arxiv.org/abs/1511.06732.

Rohrbach, A., Hendricks, L. A., Burns, K., Darrell, T., and Saenko, K. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4035–4045, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/

v1/D18-1437. URL https://www.aclweb.org/anthology/D18-1437.

Rossiello, G., Basile, P., and Semeraro, G. 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*, pp. 12–21, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1003. URL https://www.aclweb.org/anthology/W17-1003.

See, A., Roller, S., Kiela, D., and Weston, J. 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)*, pp. 1702–1723, 2019. doi: 10.18653/v1/n19-1170. URL https://doi.org/10.18653/v1/n19-1170.

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

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

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, L. and Ling, W. 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*, pp. 47–57, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1007. URL https://www.aclweb.org/anthology/N16-1007.

West, P., Holtzman, A., Buys, J., and Choi, Y. 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)*, pp. 3750–3759, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1389. URL https://www.aclweb.org/anthology/D19-1389.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Wu, L., Xia, Y., Tian, F., Zhao, L., Qin, T., Lai, J., and Liu, T. Adversarial neural machine translation. In *Proceedings of The 10th Asian Conference on Machine Learning, ACML 2018, Beijing, China, November 14-16, 2018*, pp. 534–549, 2018. URL http://proceedings.mlr.press/v95/wu18a.html.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., and Berg-Kirkpatrick, T. 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*, pp. 7298–7309, 2018.

Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., and Tenenbaum, J. 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*, pp. 1039–1050, 2018.

Yuan, L., Tay, F. E., Li, P., Zhou, L., and Feng, J. Cycle-sum: cycle-consistent adversarial lstm networks for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 9143–9150, 2019.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., and Choi, Y. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pp. 9051–9062, 2019.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Zheng, H. and Lapata, M. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6236–6247, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1628.

A. Datasets

B. Reproducibility

C. Generated Examples

D. Inferred Control Tokens

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 gave this place a shot per other Yelp reviews. I had the Capistrami 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. 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.

E. Training and Inference Speed

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
Cafés:	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

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

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	
Home Decor: encore, furniture, ikea, tint, store, ross, storage, mattress, gifts, stuff	
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	

Figure 9. Examples of Inferred control tokens for each category of venues for the Yelp dataset.

F. 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 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.

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.

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 discarded. After filtering out the “negatives” HITs and the ones from “dubious” answers, we were left with 446 annotations. We further discarded all annotations made in less than a minute to keep 377 realistic answers.

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.

set	unfiltered	negatives discarded	dubious discarded	1min discarded	agreement
average time to answer	2min16s	2min17	2min9	2min26	2min10
theoretical hourly wage	13.22	13.16	13.96	12.36	13.87

Table 5.