PASS: Perturb-and-Select Summarizer for Product Reviews

Nadav Oved *
Technion - Israel Institute of Technology
Haifa, Israel
nadavo@campus.technion.ac.il

Ran Levy
Amazon
Tel Aviv, Israel
ranlevy@amazon.com

Abstract

The product reviews summarization task aims to automatically produce a short summary for a set of reviews of a given product. Such summaries are expected to aggregate a range of different opinions in a concise, coherent and informative manner. This challenging task gives rise to two shortcomings in existing work. First, summarizers tend to favor generic content that appears in reviews for many different products, resulting in template-like, less informative summaries. Second, as reviewers often disagree on the pros and cons of a given product, summarizers sometimes yield inconsistent, self-contradicting summaries. We propose the PASS system (Perturb-and-Select Summarizer) that employs a large pre-trained Transformer-based model (T5 in our case), which follows a few-shot fine-tuning scheme. A key component of the PASS system relies on applying systematic perturbations to the model's input during inference, which allows it to generate multiple different summaries per product. We develop a method for ranking these summaries according to desired criteria, coherence in our case, enabling our system to almost entirely avoid the problem of self-contradiction. We compare our system against strong baselines on publicly available datasets, and show that it produces summaries which are more informative, diverse and coherent.[1]

1 Introduction

Online shopping has become a popular form of purchasing goods even before the most recent acceleration due to the COVID-19 pandemic. As e-commerce websites strive to make the shopping process more useful and enjoyable for customers, many interesting challenges arise. One challenge deals with how to surface opinions from product reviews in a concise yet reliable fashion. The research community has addressed this challenge early on, starting from the work of (Hu and Liu, 2004) which defined the task of mining and summarizing customer reviews. More recent advancements have relied on modern deep learning models trained on large collections of unannotated customer reviews (Brazinskas et al., 2020b,a).

Our first observation relates to the summaries generated by CopyCat (Brazinskas et al., 2020b) and FewSum (Brazinskas et al., 2020a), two of these SOTA systems, which tend to mix generic statements such as *"Would recommend this product to anyone"* along with more informative content such as *"The sound quality is good"* (see Table 6 in the Appendix for examples of such generated summaries). Due to the emphasis of summarization systems on conciseness, we maintain that generic content should be used sparingly. Additionally, even if the content is not extremely generic, customers may perceive summaries as less useful if they tend to repeat themselves across products. In order to estimate the similarity between summaries generated for different products, we devise the Set-Pairwise-ROUGE metric (henceforth denoted as *SPR*), that computes the average ROUGE (Lin, 2004b) scores of summaries for two different products, across all product pairs. Using this metric we show that human written reference summaries are indeed far more diverse than their system generated counterparts, i.e. the SPR of reference summaries is significantly lower. We henceforth denote the notion of cross product diversity of summaries as *CP-Diversity*.

Large pre-trained Transformer-based (Vaswani et al., 2017) models such as OpenAI's GPT-3 (Brown et al., 2020), Google's T5 (Raffel et al., 2020), PEGASUS (Zhang et al., 2020a), and Facebook's BART (Lewis et al., 2020) have made com-

*Completed during an internship at Amazon.

[1]Summaries generated by PASS are available at: https://registry.opendata.aws/

pelling advancements on a host of NLG tasks, including abstractive text summarization. In this work we wish to leverage such models for product reviews summarization, aiming to generally improve the quality of generated summaries, and specifically in terms of their diversity across different products. While we aim to generate human-like texts, care has to be taken with respect to their correctness. Indeed, concerns have been raised regarding the factual consistency of abstractive summaries, i.e., whether the facts conveyed in the summary agree with the source text (Cao et al., 2018; Kryscinski et al., 2019; Maynez et al., 2020).

Our second observation relates to this issue of factual consistency in the context of product reviews summarization. Our task not only faces the risk of models hallucinating incorrect information, as in traditional abstractive text summarization, but also the risk of generating self-contradicting summaries which are not caused by model hallucinations. The latter can occur when the source documents contradict one another. This situation is quite likely because reviews may disagree on some product aspects or even disagree entirely. For example, review A states a machine is *"easy to operate"* vs. review B which states it *"requires trial and error"* (see more examples in Table 7 in the Appendix). In this unique setup, factual consistency is undefined and instead we wish to measure a different characteristic: the self-consistency of the summary. To the best of our knowledge this issue has not been analyzed in the past and in some sense it renders the task ill-defined because it's not clear whether the summary is supposed to convey a range of possibly contradicting opinions about the product or the majority opinion. From here on, we shall assume that a summary has to convey the majority opinion of the reviews and do so in a self-consistent manner.

Our proposed method starts by fine-tuning a strong pre-trained language model for product reviews summarization in a few-shot setup. We then employ an input perturbation method that drops k reviews out of the input and concatenates the remaining reviews in random order. This process, denoted as *LkO*, short for *leave k out*, produces notable variation between candidate summaries, which increases the model's output diversity.[2] Once we have produced a set of candidate summaries, we essentially cast our original summary generation problem as a ranking problem. This approach gives us the choice over what kind of summary we are interested in as the final output, i.e. choosing our ranking criteria. As mentioned above, our main concern in this work is producing self-consistent summaries. Instead of basing our ranking solely on this criterion, we train a more general coherence summary ranker using human annotated coherence scores (Fabbri et al., 2021). Finally, for each product, we select the top ranked summary as the system's output.

We compare our method against strong baselines, comprised of systems introduced in previous work on multi-document opinion summarization, and a T5 language model fine-tuned for abstractive text summarization. We evaluate each over 3 dimensions, of which relevance and coherence are commonly used in summarization (Dang, 2005), and our newly introduced metric for CP-Diversity. We demonstrate that our method produces high quality summaries which are more informative, diverse and coherent.

In summary, the main contributions of this work are: (1) highlight two shortcomings of existing product reviews summarizers, namely low CP-Diversity and self-inconsistency, and propose a dedicated metric for the former. (2) Propose a method that leverages strong pre-trained models that improve the CP-Diversity while significantly reducing the risk of self-inconsistencies.

2 Related Work

Product Review Summarization. Product review summarization is a form of multi-document summarization in which a set of product reviews for a single product serves as the document cluster to be summarized. A common approach for product review summarization, which centers the summary around a set of extracted aspects and their respective sentiment, is termed *aspect-based summarization* (Hu and Liu, 2004; Kansal and Toshniwal, 2014; Wu et al., 2016; Angelidis and Lapata, 2018; Coavoux et al., 2019).

As in traditional summarization, there are two inherently different requirements for the task, a simplified one, in which the goal is to provide an extractive output, i.e., a list of sentences extracted from the review set, or a more advanced one, in which the goal is to provide an abstrac-

[2] Diversity here is between candidate summaries for the same product, not to be confused with CP-Diversity.

tive output, i.e., generated content not restricted to use the same wording of the source set. Extractive summarization include earlier works such as (Carenini et al., 2006; Lerman et al., 2009; Xiong and Litman, 2014). More recently, (Tan et al., 2017) suggested a novel generative topic aspect sentiment model, while (Angelidis et al., 2021) suggested a novel system able to extract both general and aspect-specific summaries. As for abstractive summarization, recent advances on pre-training neural networks were explored in the context of product reviews in unsupervised and few-shot learning schemes which led to promising results (Chu and Liu, 2019; Brazinskas et al., 2020b,a; Suhara et al., 2020; Amplayo et al., 2021).

Evaluating Summarization Systems. Evaluation of summarization systems is usually performed utilizing a mix of automatic metrics and human ratings. Among the automated metrics, probably the most well-known is the ROUGE family of scores (Lin, 2004b) that measures n-gram overlap between generated summaries and corresponding reference summaries. Many other metrics that aim to quantify how well generated summaries align with reference summaries have been proposed, such as BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), ROUGE-WE (Ng and Abrecht, 2015) and BertScore (Zhang et al., 2020b) to name a few. Unfortunately, such metrics alone do not tell the whole story and recently several works observed that a new requirement is necessary in order to ensure that facts from the summary agree with the source document (Cao et al., 2018; Kryscinski et al., 2019; Maynez et al., 2020). This requirement is usually known as factual consistency. As for human ratings, those are usually obtained across several dimensions of summary quality. The DUC 2005 task (Dang, 2005) suggested the following 5 dimensions: Grammaticality, Non-redundancy, Referential clarity, Focus and Structure, and Coherence.

In the context of product reviews summarization (Brazinskas et al., 2020a) use the standard ROUGE-1/2/L metrics as well human comparative judgments on 5 dimensions: Fluency, Coherence, Non-Redundancy, Informativeness and Sentiment. To the best of our knowledge the issues of self-consistency and diversity across products were not directly analyzed before.

3 Perturb-and-Select Summarizer

In this section, we propose a system that employs a large pre-trained Transformer-based model (T5) in a few-shot fine-tuning scheme for multiple reviews abstractive summarization. We aim to leverage the inherent diversity between reviews for a given product to our advantage, by applying systematic perturbations to the model's input during inference. This allows our fine-tuned model to generate multiple different candidate summaries per product, exhibiting variability both in the content being surfaced as well as in the phrasing of said content. We develop a ranking mechanism for selecting the best candidate summary according to desired criteria, which in our case is coherence. We provide an end-to-end diagram of the PASS Summarizer's components in Figure 1.

3.1 Fine-tuning T5 for Summary Generation

PASS relies on a pre-trained T5 language model, which we fine-tuned on a small publicly available dataset for product reviews summarization (Brazinskas et al., 2020a). We follow a similar fine-tuning scheme for abstractive text summarization to the one presented in (Raffel et al., 2020) with the exception that we concatenate the multiple reviews into a single input text as a pre-processing step. As the dataset contains multiple reference summaries per product, we repeat our training process for each reference summary using the same (concatenated) input text.

3.2 Candidate Summary Generation

In light of the natural diversity existing between product reviews, we explore a modeling approach which allows for such diversity to emerge in our summarizer's output as well. We do this by manipulating the model's input, sampling which reviews to use each time, in a way that allows for increasing the relative prevalence of certain reviews over others. We also re-shuffle the reviews before concatenation to ensure the model is not affected by their internal order. Note that prior attempts have been made to directly manipulate the content within the reviews (Amplayo and Lapata, 2020) a path that we do not explore here. Our intervention method guarantees that each review's correctness, integrity and meaning are preserved. Since it only affects the subset of reviews being used and their order of concatenation, this increases the potential for diversity (per product and across products)

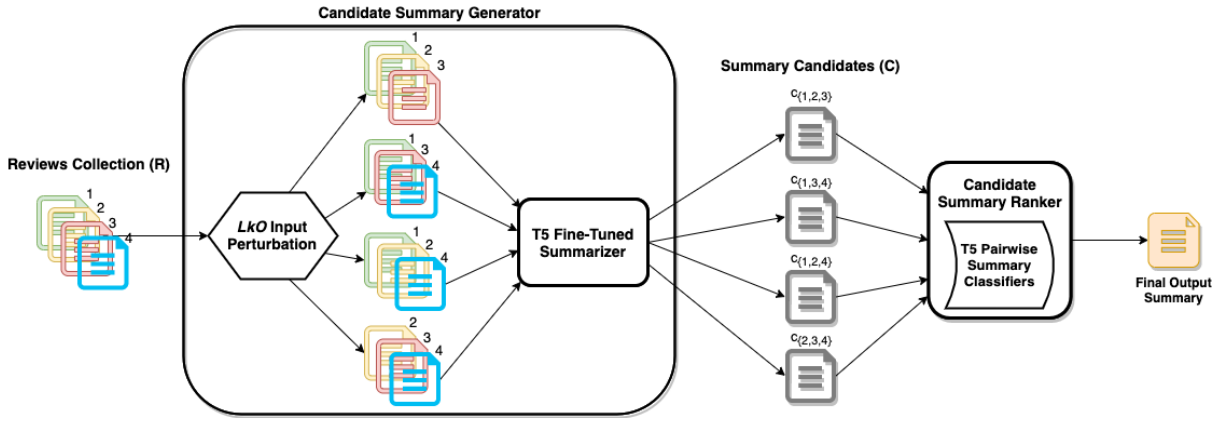

Figure 1: A diagram of the PASS components, with an example for a collection of reviews of size $d = 4$, $k = 1$.

emerging from the input's content, without compromising its linguistic quality.

***LkO* Input Perturbation Method.** Given a set of d reviews $R = \{r_1, ..., r_d\}$ for a product p, our perturbation method iterates over $A(R)$ the set of all possible subests of size $d - k$ in R, $A(R) = \{S | S \subset R, |S| = d - k, 1 \leq k < d\}$. Given a subset $S \in A(R)$ we concatenate its reviews in random order, and feed the concatenated text into our fine-tuned T5 summarizer, which generates a candidate summary c. We repeat this step for all $S \in A(R)$, resulting in a set of generated candidate summaries which we denote as $C = \{c_1, ..., c_m\}, m = \binom{d}{k}$. This process, denoted as *LkO*, short for *leave-k-out*, produces notable variation between candidate summaries (see Table 8 in the Appendix for examples), and allows for different content and aspects to emerge in the summaries, which were less likely to have surfaced otherwise. We found that this perturbation approach produces higher variation across candidate summaries when applying it on the model's input only during the inference stage, not during training. Our method produces multiple perturbed versions of a given input while its references remain the same. If applied during training, this might encourage the model to fit a larger range of input features to a smaller set of outputs. We are interested in the opposite effect - we would like to encourage higher output variation as a function of input diversity.

Note that when dealing with large review sets, achieving diversity does not require iterating over all subsets in $A(R)$. For such scenarios, we recommend constructing a fixed number (m) of randomly sampled review subsets, so long as m is sufficiently large. In our experiments we employ the full *LkO* input perturbation method, since standard datasets focus on relatively small review sets.[3]

An alternative method for increasing novelty and variability in the output of a generative language model, is to directly intervene in its decoding algorithm, e.g., Beam Search (Vijayakumar et al., 2016; Cibils et al., 2018). Note that this will not have the same effect as our proposed approach. First, since beam search is a decoding algorithm, it only has access to the underlying language model, and is completely separated from the model's input. Second, beam search's mechanism is fixed to make local word-by-word decisions, before the complete summary is revealed. Finally, our approach guarantees that given a set of input texts, at least one candidate output will not be influenced at all by a specific input text (or more if $k > 1$). For example, if a set of 4 reviews contains 3 reviews discussing price, and 1 review discussing quality, our method guarantees that at least 1 candidate summary will be generated solely based on the first three (discussing price). Furthermore, our method increases the probability for a summary to mention both price and quality, when a review discussing price is left out.

3.3 Candidate Summary Ranking

Once a set of candidate summaries are generated per product, we have essentially cast our summary generation problem as a summary ranking problem. This allows us to retrieve a summary, which ranks best out of a diverse set of candidates, according to desired, interpretable criteria.

[3] A few recent works attempt to explicitly address this issue (Shapira and Levy, 2020; Angelidis et al., 2021).

As mentioned in Section 1, our main concern is producing CP-diverse yet self-consistent and coherent summaries. Since our input perturbation method generates multiple candidate summaries, we are now left with the task of ranking this set by coherence. We would like the ranking process to filter out self-contradicting, incoherent or inconsistent candidates (by assigning low rank) and to promote well-formed, coherent candidates to the top of the list. To achieve this, we train a classifier that receives two summaries as input and decides whether the first summary is more coherent than the second or the opposite. The classifier can also decide that both summaries are equally coherent. Using such a classifier, we can obtain a partial ranking of the reviews by running all pairwise comparisons and count the number of times each summary was better than the summary it was paired with.

Pairwise Summary Classifier. We train a model to classify a pair of summaries for coherence, by fine-tuning a pre-trained T5 model for pairwise text classification. Given a pair of summaries, the model is required to classify them as either: summary A is more coherent, summary B is more coherent, or A and B are equivalent in terms of coherence. A pair of summaries can often be considered equivalent when judging them according to specific criteria, stemming from the natural fact that often more than one summary can be considered correct or good. Indeed it has been shown that several reference summaries are needed for reliable evaluation showing that there is more than one truth (Lin, 2004a). Since this model is used as a comparator for ranking candidate summaries, we are especially sensitive to specific types of classification errors. If the model mistakenly classifies a summary to be more coherent than the other while the opposite is true, we consider this a critical classification error. This type of error could be detrimental to the validity of the ranking process, therefore we aim to minimize its rate. While other types of errors also reduce the classifier's accuracy, we consider a mistake where the model classifies two summaries to be equivalent when in truth one is more coherent than the other, as less harmful for ranking purposes.

Ranking Method. Our proposed ranking method iterates over all possible pairs of candidate summaries for a given product, and counts how many times each candidate was classified by the coherence pairwise classifier (our primary comparator), as more coherent than its counterpart. As a tie-breaking, secondary comparator, we train an additional pairwise summary classifier, to classify which candidate is more fluent, out of a pair of given candidates. We select the top ranked candidate as the final output summary for each product.

4 Experimental Setup

4.1 Data

We utilize a recent publicly available Amazon product reviews summarization dataset (Brazinskas et al., 2020a) for fine-tuning the T5 model which underlines the *PASS* system and for evaluating the *LkO* input perturbation method, both in isolation and as part of the end-to-end *PASS* system. The dataset contains product reviews and reference summaries for 60 products on Amazon. Each product has 8 reviews and 3 reference summaries written by crowd source workers. We follow the dataset splits to the training, development and test sets provided by the authors of the dataset. While we mainly focus on product reviews summarization, we include the Yelp business reviews summarization dataset (also from (Brazinskas et al., 2020a)) in our end-to-end evaluation for the sake of completeness. The Yelp dataset contains business reviews and reference summaries for 100 businesses.

For training and evaluating the pairwise coherence classifier, we utilize a public dataset of human annotated summaries (Fabbri et al., 2021), generated by 16 modern text summarization models for 100 news articles (1600 examples in total) from the CNN/DailyMail dataset (Hermann et al., 2015). Each summary was rated (on a scale of 1 to 5) across 4 dimensions: coherence, consistency, fluency and relevance, by 5 independent crowd source workers and 3 independent experts (8 annotations in total). We chose to use the experts' annotations only, as they are considered to be more accurate and reliable for coherence and fluency (Fabbri et al., 2021). We construct a pairwise version of this dataset, by creating summary pairs from all 16 model outputs for each of the 100 news stories, along with their annotation scores for each metric respectively. We split the dataset according to news stories, by randomly sampling 20 stories for the test set, 16 stories for the develop-

ment set and the rest are used for the training set. Given a pair of summaries (a, b), their respective average expert rating, (r_a, r_b) and a threshold parameter ϵ, we define the label for that pair as:

$$label(a,b) = \begin{cases} A, & \text{if } r_a - r_b \geq \epsilon \\ B, & \text{if } r_b - r_a \geq \epsilon \\ E, & \text{otherwise} \end{cases}$$

where E denotes the case where both summaries are equivalent, A denotes that summary a is better than b and B denotes the opposite. To ensure that our training data is invariant to a pair's internal order, we create examples for all (a, b) and (b, a) pairs in the training set.

4.2 Experimental Details

Fine-tuning T5 for Summary Generation. We fine-tune a T5-Base model ($220M$ parameters (Raffel et al., 2020)) for abstractive text summarization as described in 3.1 on the training set, and tune its hyperparameters on the development set. We train for maximum 20 epochs while employing a standard early stopping mechanism (Falcon, 2019) based on the development set's average loss per epoch. We fine-tune a separate model for the Amazon and Yelp datasets. Hyperparameters and further details can be found in Section 7.1 in the Appendix.

***LkO* Input Perturbation.** We experiment with the *LkO* method described in Section 3.2 with $k \in \{1, 2, 3, 4, 5\}$ on the development set. For the end-to-end system we choose $k = 2$ aiming to obtain high output diversity while limiting computation complexity, and avoiding the risk of dropping a majority of the reviews ($k > 4$) each time. We provide evaluation details in 5.1.

Pairwise Summary Classifier. We train two T5-Base models to classify which summary is better, one in terms of coherence, to be used as our ranking method's primary comparator, and one in terms of fluency to break ties. We experimented with different values for $\epsilon \in \{0.25, 0.5, 0.75, 1.0\}$, and chose $\epsilon = 0.5$ for the coherence classifier and $\epsilon = 0.25$ for the fluency classifier. The choice of ϵ was based on dataset statistics per metric and evaluation of each model's performance on the development set.

Baselines. We compare the PASS system to four baselines:

COPYCAT (Brazinskas et al., 2020b) is an unsupervised reviews summarizer that is trained to generate a review given other reviews for the same product. The authors suggest a novelty mechanism that controls the extent to which the summary deviates from the inputs.

FEWSUM (Brazinskas et al., 2020a) is a few-shot reviews summarizer that builds upon the ideas of CopyCat but also conditions the model on certain linguistic properties such as writing style.

T5 is the pre-trained T5-base language model which was not fine-tuned. We do not report results for this model, as it consistently performed worst.

T5-FT is the fine-tuned T5-base model described above.

We do not report results for MEANSUM (Chu and Liu, 2019) since it was consistently outperformed by FEWSUM (Brazinskas et al., 2020a).

5 Evaluation

5.1 Candidate Summary Generation

Recall that our main objective for generating candidate summaries is to encourage output diversity. Hence, we would like to verify that our perturbation method, *LkO*, produces sufficiently diverse candidates for a given product. In order to measure textual diversity between candidate summaries for a given product, we need to devise a diversity metric. We propose the SPR metric (shorthand for Set-Pairwise-ROUGE) which measures the opposite of diversity, i.e., the average lexical similarity across pairs of summaries from a given set. We base SPR on ROUGE F1 scores for any n-gram level, therefore SPR-1 relies on ROUGE-1 F1 scores and so on.

SPR Formal Definition. For a given set of summaries $S = \{s_1, ..., s_n\}$, we define the set of all pairs from S as $P(S) = \big\{\{s_i, s_j\} \big| s_i \in S, s_j \in S, i \neq j\big\}$. We then define the set-pairwise-rouge (SPR) metric as:

$$SPR(S) = \frac{1}{|P(S)|} \cdot \sum_{\{s_i,s_j\} \in P(S)} ROUGE(s_i, s_j)$$

Note that SPR is a general metric of diversity, applicable to an arbitrary set of summaries. Therefore, it can be applied to measure both IP-Diversity (in-product diversity, as we do here) and CP-Diversity (cross-product diversity, as we do in Section 5.3). For clarity, we shall denote IP-SPR when measuring IP-Diversity and CP-SPR when measuring CP-Diversity with SPR.

Dataset	System	Length	R-1	R-2	R-L	CP-SPR-1	CP-SPR-2	CP-SPR-L	Coherence
Amazon	CopyCat	33.45	27.85	4.77	18.86	36.29	14.12	29.52	–
	FewSum	52.50	33.56	7.16	21.49	34.54	10.61	23.93	-0.200
	T5-FT	52.75	37.07	**9.68**	**23.47**	**25.56**	3.32	**17.38**	-0.050
	PASS	47.75	**37.43**	8.02	23.34	25.79	**2.63**	**17.38**	**0.150**
	Gold	49.82	–	–	–	19.48	1.61	13.00	0.100
Yelp	FewSum	52.9	37.29	9.92	22.76	40.82	17.09	30.34	0.050
	T5-FT	40.58	**38.72**	**10.26**	**24.47**	38.93	13.05	29.55	-0.250
	PASS	52.15	36.91	8.12	23.09	**30.88**	**6.35**	**21.33**	**0.200**
	Gold	49.81	–	–	–	24.41	2.80	15.98	0.000

Table 1: End-to-End results on the Amazon (top) and Yelp (bottom) test sets. **R** stands for average ROUGE F1 scores with reference summaries, **CP-SPR** for Set-Pairwise-ROUGE scores measuring CP-Diversity and **Coherence** for Best-Worst Scaling scores, which range from -1 (unanimously worst) to +1 (unanimously best), on a crowdsourced human evaluation task.

Figure 2 depicts a box plot of the IP-SPR-2 scores for k ranging from 1 to 5. We observe the biggest drop in similarity (increase in diversity) between $k = 1$ and $k = 2$. While we aim to increase diversity, we are also mindful of the increase in runtime as k grows. Additionally, we would like to avoid sampling out a majority of reviews ($k > 4$), since the risk of generating a summary with minority view or low informativeness also increases with k. Indeed, as shown in Figure 3, which depicts a similar box plot but this time of the ROUGE-2 scores against the reference summaries, the variance increases with k and the worst-case ROUGE-2 score decreases with k.

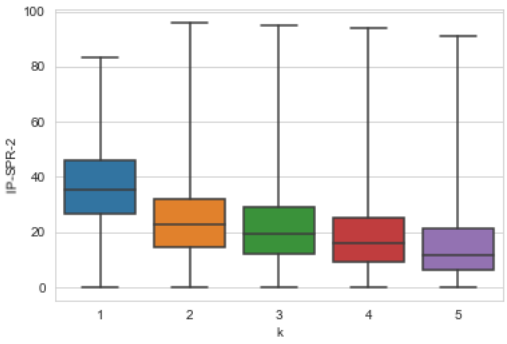

Figure 2: IP-SPR-2 scores (measuring IP-Diversity) box plot, for all pairs of candidate summaries generated with *LkO* input perturbation method for $k = 1, ..., 5$.

While diversity is certainly not the only aspect for evaluating generated summaries, we explore other dimensions in the following sections.

5.2 Candidate Summary Ranking

The pairwise summary classifiers can be evaluated directly using human scores from (Fabbri et al., 2021) after adapting them to our ternary classification task. Figure 4 depicts the confusion matrix for our coherence classifier. We observe that the estimated probability of a critical error (choosing A over B or B over A) is very low, 0.05, while at the same time the overall accuracy of 0.61 is reasonably high compared to 0.33 and 0.36 achieved by the random and majority (always predicts that A and B are equally coherent) baselines respectively. Applying the classifier to a set of 28 candidates per product, yields a single top ranking candidate for 70% of products in the Amazon test set.

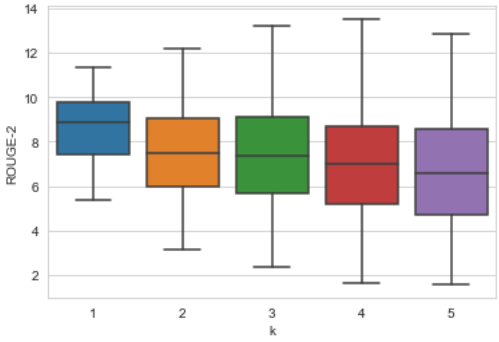

Figure 3: ROUGE-2 F1 scores box plot, for all candidate summary sets generated with *LkO* input perturbation method for $k = 1, ..., 5$.

To further break ties, we utilize the fluency classifier as a secondary comparator. See Figure 10 in the appendix for a similar confusion matrix for the fluency classifier. Again, the probability for a critical error is very low, 0.0125, while the overall accuracy is 0.67. After applying fluency as a tie breaker, we find that all products in the Amazon test set have a unique top ranking summary.

The training data for both classifiers comes

from a domain (News Articles) which is different from our main dataset's domain (Product Reviews). We hypothesize that coherence and fluency are linguistic properties that are not heavily tied with the domain, since they relate to a summary's overall collective and individual sentence quality (Dang, 2005). Indeed, our results show (see Table 2) that PASS benefited from this data despite the risk of a possible domain shift.[4]

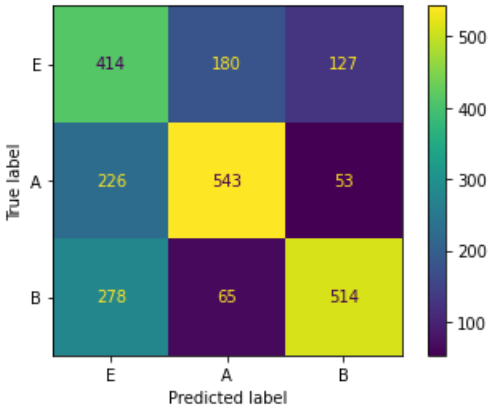

Figure 4: Confusion matrix for the Coherence Pairwise Classifier.

5.3 End-to-End System

We evaluate our end-to-end system across 3 dimensions. The first, informativeness, is traditionally evaluated using the ROUGE-1/2/L F1 measures (Lin, 2004b) and we follow suit. The second dimension, which subsumes the self-consistency issue, is coherence. To this end, we conducted a crowdsourced human evaluation task, which compares between the generated summaries of 4 different summarization systems, including our proposed PASS system. We used Best-Worst Scaling (Louviere and Woodworth, 1991; Louviere et al., 2015; Kiritchenko and Mohammad, 2016, 2017) to compute each system's score as the difference between the percentage of times it was selected as *best*, and the percentage of times it was selected as *worst* (Orme, 2009). This is inline with prior work on product review summarization (Brazinskas et al., 2020b,a). As for our third dimension, recall that we would like our system to generate diverse summaries across different products, a notion that we denoted as CP-Diversity. Lacking an existing metric, we use our previously defined SPR-1/2/L measure on the set of final (top-ranked) summaries across all test set products.

Table 1 reports results for all 3 dimensions. For the Amazon dataset (top table), we observe that *PASS* outperforms the baselines in coherence and CP-Diversity while keeping a comparable informativeness to the next best system, *T5-FT*. The only exception being ROUGE-2 in which *T5-FT* outperforms *PASS* which could be explained by the somewhat longer summaries it generates. Interestingly, in CP-Diversity, the performance of *PASS* is closer to human performance than to *CopyCat* and *FewSum* but there's still room to make the summaries even more diverse. For the sake of completeness and following previous work (Chu and Liu, 2019; Brazinskas et al., 2020b,a) we report results on business reviews from the Yelp dataset in the bottom of Table 1.

Recall that our key goals were to avoid generating summaries containing crude coherence (CE) and self-consistency (SCE) errors (see Table 3 for examples of such errors). In order to evaluate these directly, both authors independently marked each of the summaries generated by *FewSum*, *T5-FT* and *PASS* for the Amazon test set as having a crude error or not, for both types of errors. Table 2 reports the ratios of crude errors per system, considering cases where at least one annotator (I) and both annotators (II) marked as crude. We measured the level of agreement between the two annotators by calculating Cohen's Kappa coefficients (Cohen, 1960) for each annotation task, which resulted in $\kappa_{CE} = 0.571$ and $\kappa_{SCE} = 0.779$.

System	CE-I	CE-II	SCE-I	SCE-II
FewSum	0.50	0.34	0.3	0.25
T5-FT	0.38	0.25	0.3	0.2
PASS	**0.19**	**0.09**	**0.05**	**0.00**

Table 2: Ratios of crude coherence (CE) and self-consistency (SCE) errors for each system on the Amazon test set. I/II refer to cases where at least one/both annotators marked the summary as having an error.

Finally, for a qualitative impression we provide in Table 4 an example of the systems' outputs for a product from the Amazon test set.

6 Conclusion

In this work we highlight two shortcomings of existing product reviews summarization systems,

[4]While we did not find evidence suggesting a domain shift, it is an aspect we leave for further investigation in future work.

Tights. *These tights are very comfortable and durable. They can be worn with ballet slippers or sandals. The color is beautiful and the fabric is soft. They will last a long time. They are great for transitioning from ballet to ballet.*
Purse. *This purse is not as cute as it looks in the picture. It is very small and will not hold a lot of stuff. It would be a great purse if it was a little bigger but it would have been nice to have a purse that would hold more than one purse.*
Protein Bar. *These bars are a great snack bar. They taste good and have a good amount of protein. They do not have a lot of protein in them so they are not as sweet as some protein bars, but for the price, they are well worth it.*
Tank Top. *This tank top is well made, fits well, and is comfortable to wear. The only thing is that it runs a little small, so order a size up from what you normally wear. Other than that, it's a great top. It's well made and it looks like it will last a long time. Love it!*

Table 3: Example of summaries generated by T5-FT and FewSum models for different products in the Amazon test set, which contain crude errors (CE) and self-consistency errors (SCE).

PASS. *These Reeboks are great for supporting a high arch and are lightweight and comfortable. They come in a variety of colors and sizes, and are ideal for walking or biking. They are also flexible and well made.*
T5-FT. *These Reeboks are a great choice for those with wide feet. They run true to size and the colors are great. They are lightweight and comfortable, yet they are flexible and flexible. They are recommended for people with wide feet. They are also very popular for running and casual wear.*
FewSum. *These running shoes are great! They fit true to size and are very comfortable to run around in. They are light weight and have great support. They run a little on the narrow side, so make sure to order a half size larger than normal.*
CopyCat. *I love these shoes. They are light weight and comfortable to wear. I have worn them for several months now and they are holding up well. I would recommend them to anyone looking for a comfortable shoe.*

Table 4: Example of summaries generated by PASS, T5-FT, FewSum and CopyCat systems for the same sports shoes reviews.

namely low CP-Diversity and self-inconsistency. We propose the SPR metric to quantify cross product similarity of summaries and demonstrate that indeed, humans summaries are far more diverse than system generated summaries. To overcome this issue we rely on stronger pre-trained models such as the recent T5 model which significantly improves the CP-Diversity. However, the second problem still remains and even intensifies as without the safety net of generic content, the risk of incoherent or even self-contradicting text is substantial. To this end, we propose the Perturb and Select summarizer (*PASS*). In the first step, *PASS* applies systematic perturbations to the input texts in a way that allows the T5 model to generate multiple summary candidates that sufficiently differ from one another. Given such a set of diverse summaries, *PASS* applies a trained ranker to smartly select a promising candidate in terms of coherence. Finally, we show that the resulting *PASS* system, outperforms SOTA models in the domain of product reviews in terms of informativeness, CP-Diversity and coherence. When comparing to a fine-tuned T5 model *PASS* outperforms it in coherence and CP-Diversity, while maintaining comparable performance for informativeness.

In future work we plan to investigate the Perturb-and-Select framework in order to promote summaries with a plethora of desired linguistic characteristics, other than coherence. We shall further explore ways of extending this framework to employ other input perturbation methods and experiment with scenarios of larger scale input. In addition, we plan to further investigate our proposed SPR evaluation metric for lexical diversity, by studying its correlation with human judgments. Lastly, we believe our proposed framework and evaluation metric may be applicable to other domains of opinion or news summarization.

Acknowledgements

We would like to thank Hila Gonen, Iftah Gamzu and anonymous reviewers, who helped improve the draft with their invaluable comments and insight.

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Unsupervised opinion summarization with content planning. In *AAAI*.

Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised opinion summarization with noising and denoising. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1934–1945, Online. Association for Computational Linguistics.

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive opinion summarization in quantized transformer spaces. *Trans. Assoc. Comput. Linguistics*, 9:277–293.

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.

Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2020a. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4119–4135. Association for Computational Linguistics.

Arthur Brazinskas, Mirella Lapata, and Ivan Titov. 2020b. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5151–5169. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4784–4791. AAAI Press.

Giuseppe Carenini, Raymond T. Ng, and Adam Pauls. 2006. Multi-document summarization of evaluative text. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics.

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*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR.

André Cibils, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2018. Diverse beam search for increased novelty in abstractive summarization. *CoRR*, abs/1802.01457.

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.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12.

Alexander R. Fabbri, Wojciech Kryscinski, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir R. Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Trans. Assoc. Comput. Linguistics*, 9:391–409.

WA Falcon. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177. ACM.

Hitesh Kansal and Durga Toshniwal. 2014. Aspect based summarization of context dependent opinion words. In *18th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, KES 2014, Gdynia, Poland, 15-17 September 2014*, volume 35 of *Procedia Computer Science*, pages 166–175. Elsevier.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Svetlana Kiritchenko and Saif M. Mohammad. 2016. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, San Diego, California.

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, Hong Kong, China. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan T. McDonald. 2009. Sentiment summarization: Evaluating and learning user preferences. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pages 514–522. The Association for Computer Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004a. Looking for a few good metrics: Automatic summarization evaluation - how many samples are enough? In *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization, NTCIR-4, National Center of Sciences, Tokyo, Japan, June 2-4, 2004*. National Institute of Informatics (NII).

Chin-Yew Lin. 2004b. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jordan J Louviere, Terry N Flynn, and Anthony Alfred John Marley. 2015. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.

Jordan J Louviere and George G Woodworth. 1991. Best-worst scaling: A model for the largest difference judgments. Technical report, Working Paper.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal. Association for Computational Linguistics.

B. Orme. 2009. Maxdiff analysis : Simple counting , individual-level logit , and hb.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ori Shapira and Ran Levy. 2020. Massive multi-document summarization of product reviews with weak supervision. *CoRR*, abs/2007.11348.

Yoshihiko Suhara, Xiaolan Wang, Stefanos Angelidis, and Wang-Chiew Tan. 2020. OpinionDigest: A simple framework for opinion summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5789–5798.

Jiaxing Tan, Alexander Kotov, Rojiar Pir Mohammadiani, and Yumei Huo. 2017. Sentence retrieval with sentiment-specific topical anchoring for review summarization. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 2323–2326. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020a. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Thomas Wolf, Quentin Lhoest, Patrick von Platen, Yacine Jernite, Mariama Drame, Julien Plu, Julien Chaumond, Clement Delangue, Clara Ma, Abhishek Thakur, Suraj Patil, Joe Davison, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angie McMillan-Major, Simon Brandeis, Sylvain Gugger, François Lagunas, Lysandre Debut, Morgan Funtowicz, Anthony Moi, Sasha Rush, Philipp Schmidd, Pierric Cistac, Victor Muštar, Jeff Boudier, and Anna Tordjmann. 2020b. Datasets. *GitHub. Note: https://github.com/huggingface/datasets*, 1.

Haibing Wu, Yiwei Gu, Shangdi Sun, and Xiaodong Gu. 2016. Aspect-based opinion summarization with convolutional neural networks. In *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, pages 3157–3163. IEEE.

Wenting Xiong and Diane Litman. 2014. Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1985–1995, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

7 Appendix

7.1 PASS Implementation Details and Hyperparameters

All models were implemented with the PyTorch (Paszke et al., 2019) deep learning framework, utilizing the T5 (Raffel et al., 2020) pre-trained model and tokenizer implementations from HuggingFace's Transformers (Wolf et al., 2020a) library, evaluation metrics from HuggingFace's Datasets (Wolf et al., 2020b) library and PyTorch Lightning (Falcon, 2019) as a model training framework.

7.1.1 T5 Fine-Tuned Summarizer

We fine-tune a pre-trained T5-Base model ($220M$ parameters (Raffel et al., 2020)) for product reviews summarization (an abstractive text summarization task) on the training set, employing the Adam optimizer (Kingma and Ba, 2015) with weight decay (Loshchilov and Hutter, 2019). We train the model for a maximum of 20 epochs on a single NVIDIA Tesla V100 GPU, while employing a standard early stopping mechanism (Falcon, 2019) based on the development set's average loss per epoch. We employ a standard beam search decoding algorithm during inference for generating text. We tune the model's hyperparameters on the development set, and provide a list of the final model's tuned hyperparametrs along with the range of values tested during tuning.

Hyperparameters

T5 Encoder

- Max input sequence length = 512 tokens
- Training batch size = 8, $[8, 12, 16]$
- Evaluation batch size = 12, $[8, 12, 16]$

Adam Optimizer

- Learning rate = $3e-4$, $[1e-4, 3e-4, 5e-4]$
- $\epsilon = 1e-8$, $[1e-8, 3e-8, 5e-8]$
- Weight decay: 0.0
- Number of warmup steps: 0
- Gradient accumulation steps = 2, $[1, 2, 4]$
- Max gradient norm = 1.0

T5 Decoder

- Max output sequence length = 128 tokens
- Min output sequence length = 16 tokens
- Beam size = 2, $[2, 3, 4]$
- Length penalty = 2, $[1, 2, 3]$
- Repetition penalty = 2, $[1, 2, 3]$

***LkO* Input Perturbation (PASS system only)**

- $k = 2$, $[1, 2, 3, 4, 5]$

7.1.2 Pairwise Summary Classifiers

For each pairwise summary classifier (coherence, fluency), we fine-tune a pre-trained T5-Base model ($220M$ parameters (Raffel et al., 2020)) for abstractive text summarization task on the respective training set employing the Adam optimizer (Kingma and Ba, 2015) with weight decay (Loshchilov and Hutter, 2019). We train for a maximum of 20 epochs on a single NVIDIA Tesla V100 GPU, while employing a standard early stopping mechanism (Falcon, 2019) based on the development set's average loss per epoch. We employ a standard greedy decoding algorithm during inference for generating the class label. We tune the model's hyperparameters on the development set, and provide a list of the final model's tuned hyperparametrs along with the range of values tested during tuning.

Hyperparameters

Dataset

- Coherence scores difference threshold $\epsilon = 0.5$, $[0.25, 0.5, 0.75, 1.0]$
- Fluency scores difference threshold $\epsilon = 0.25$, $[0.25, 0.5, 0.75, 1.0]$

T5 Encoder

- Max input sequence length = 400 tokens
- Training batch size = 16, $[8, 12, 16]$
- Evaluation batch size = 16, $[8, 12, 16]$

Adam Optimizer

- Learning rate = $1e-4$, $[1e-4, 3e-4, 5e-4]$
- $\epsilon = 1e-8$, $[1e-8, 3e-8, 5e-8]$
- Weight decay: 0.0
- Number of warmup steps: 0
- Gradient accumulation steps = 4, $[1, 2, 4]$
- Max gradient norm = 1.0

T5 Decoder

- Max output sequence length = 2 tokens
- Min output sequence length = 2 tokens

7.2 Summary Examples

We provide examples for output summaries generated by the different summarization systems discussed in the main paper. Each example qualitatively highlights a different aspect by which we evaluate the quality of a summary, or identify its shortcomings.

PASS. *This camera is good to have as a first camera before investing in a DSLR. The quality of the pictures is great, and the camera is easy to use. It takes some time to learn about the features and settings, but overall it's a great camera.*
T5-FT. *This camera is a great camera for taking professional photos. It is easy to use and takes excellent pictures. The low light feature is outstanding and will be helpful in museums and other venues where flash is not allowed. The battery is constantly malfunctioning making the camera unusable. The on off button is also malfinctioning.*
FewSum. *This camera is a great camera for the price. It takes great pictures and is easy to use. The only drawback is that the battery life is not as good as the camera that comes with the camera. It would be nice if it had a battery life to last longer. Overall, it's a good camera.*
CopyCat. *This is a great camera for the price. It is easy to set up and use. The only downside is that it takes a while to learn how to use it, but it's not a problem.*

Table 5: Example of summaries generated by PASS, T5-FT (Raffel et al., 2020), FewSum (Brazinskas et al., 2020a) and CopyCat (Brazinskas et al., 2020b) systems for the same reviews for a digital camera.

Travel Sound Conditioner. *This is a great product for the price. The sound quality is good and the sound is good. The only problem is that it is not loud enough for a small room. It is loud enough to drown out background noise, but not very loud. Overall, it's a good product and would recommend it to anyone.*
Motion Sickness Tablets. *This is a great product at a great price. It is easy to use and easy to take. The pills are easy to swallow and do not take up a lot of space. The price is great for a product that will last a long time. Would recommend this product to anyone who suffers from nausea or sickness.*
Digital Camera. *This camera is a great camera for the price. It takes great pictures and is easy to use. The only drawback is that the battery life is not as good as the camera that comes with the camera. It would be nice if it had a battery life to last longer. Overall, it's a good camera.*

Table 6: Example of similar summaries generated by FewSum (Brazinskas et al., 2020a) for three different products.

Review 1. *The machine is very tricky. It requires some trial and errors to make it work right. I do not like to put oil in the dough; however, it appears to me that without oil it is impossible to make tortilla or chapatti. It is useless for me.*
Review 2. *Fun and easy to use! Took me one batch to get my technique worked out, but it was very simple, easy to follow directions. Easy clean up too! I would recommend this to anyone looking for an electric tortilla maker!*
Summary. *This tortilla maker is a great option for making tortillas but it does require some trial and error to make it work right. It requires some trial and error to make it work right. Yes, you should grill them after cooking to get the toasted look. It is easy to use and very easy to clean up.*

Table 7: Example of a self-contradicting summary generated by our fine-tuned T5 (T5-FT) model.

Candidate 1. *These NuGo bars are high quality and they come in a variety of flavors and sizes which make them perfect for serving as a snack or as a replacement for processed foods. They are low glycemic and have a smooth, vanilla-like texture which makes them very good.*
Candidate 2. *These NuGo bars are high quality and they come in a variety of flavors and sizes which makes them ideal for snacking on the go. The taste is great and the nutritional value is great as well. Although they can be a little sweet, they are not too sweet.*
Candidate 3. *These NuGo bars are high quality and they come in a variety of flavors and sizes. They are low glycemic and have a great taste. While they may be sweet, they can also have a chalky or barky texture. These are great for replacing junk food with healthy snacks.*
Candidate 4. *These NuGo bars are high quality and they taste great. They are low glycemic, and they contain no added sugar or artificial flavors. These are great for a healthy snack or for a quick breakfast.*
Candidate 5. *These NuGo bars are very good quality and they come in a variety of flavors. They are high in calories and fiber, and are great for snacking on the go. They are often a bit chewy, but they are definitely worth the money.*

Table 8: Example of 5 candidate summaries (out of 28) generated by PASS for the same product with *L2O* input perturbation.

7.3 Evaluation Figures

We provide figures which extend those appearing in the Evaluation section of the main paper.

7.3.1 Candidate Summary Generation

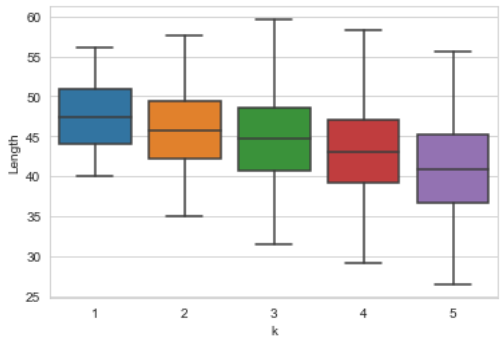

Figure 5: Length box plot for all candidate summary sets generated with *LkO* input perturbation method for $k = 1, .., 5$.

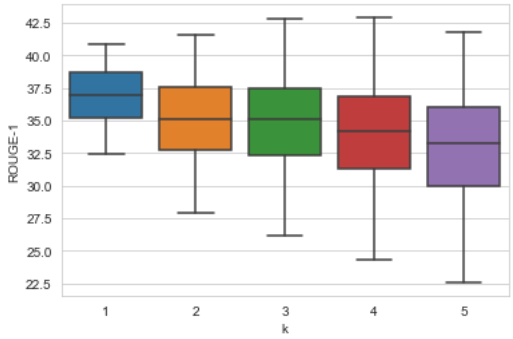

Figure 6: ROUGE-1 box plot for all candidate summary sets generated with *LkO* input perturbation method for $k = 1, .., 5$.

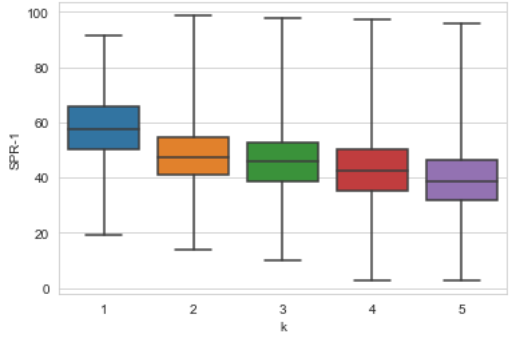

Figure 7: SPR-1 box plot for all pairs of candidate summaries generated with *LkO* input perturbation method for $k = 1, .., 5$.

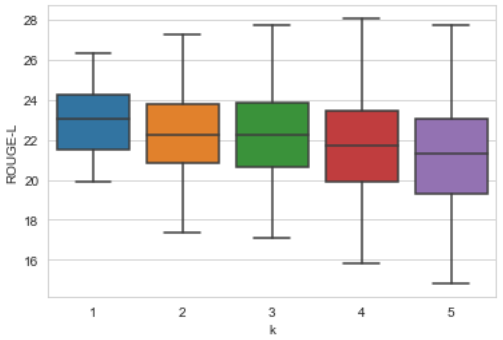

Figure 8: ROUGE-L box plot for all candidate summary sets generated with *LkO* input perturbation method for $k = 1, .., 5$.

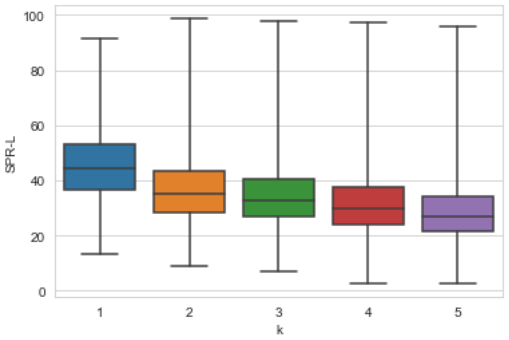

Figure 9: SPR-L box plot for all pairs of candidate summaries generated with *LkO* input perturbation method for $k = 1, .., 5$.

7.3.2 Candidate Summary Ranking

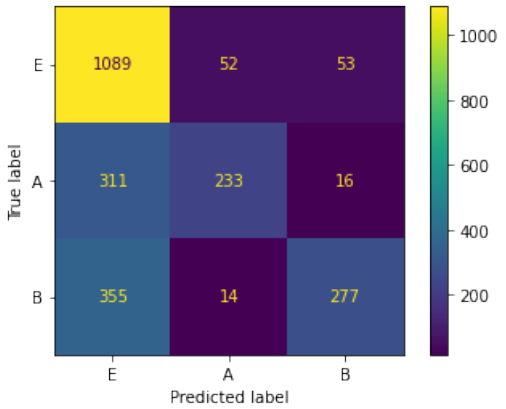

Figure 10: Confusion matrix for the Fluency Pairwise Classiifer.