Transductive Learning for Abstractive News Summarization

Anonymous ACL submission

Abstract

Pre-trained and fine-tuned news summarizers are expected to generalize to news articles unseen in the fine-tuning (training) phase. However, these articles often contain specifics, such as events and people, a summarizer could not learn about in training. This applies to scenarios such as when a news publisher trains a summarizer on dated news and wants to summarize incoming recent news. In this work, we explore the first application of *transductive learning* to summarization where we further fine-tune models on test set's input. Specifically, we construct references for learning from article salient sentences and condition on the randomly masked articles. We show that this approach is also beneficial in the fine-tuning phase when extractive references are jointly predicted with abstractive ones in the training set. In general, extractive references are inexpensive to produce as they are automatically created without human effort. We show that our approach yields state-of-the-art results on CNN/DM and NYT datasets, for instance, more than 1 ROUGE-L points improvement on the former. Moreover, we show the benefits of transduction from dated to more recent CNN news. Finally, through human and automatic evaluation, we demonstrate improvements in summary abstractiveness and coherence.

Abstractive	The penalty is more than 10 times the previous record, according to a newspaper report. Utility commission to force Pacific Gas & Electric Co. to make infrastructure improvements. Company apologizes for explosion that killed 8, says it is using lessons learned to improve safety.
Extractive	The California Public Utilities Commission on Thursday said it is ordering Pacific Gas & Electric Co. to pay a record $1.6 billion penalty for unsafe operation of its gas transmission system, including the pipeline rupture that killed eight people in San Bruno in September 2010. Most of the penalty amounts to forced spending on improving pipeline safety. On September 9, 2010, a section of PG&E pipeline exploded in San Bruno, killing eight people and injuring more than 50 others.
Ours	Pacific Gas & Electric Co. is ordered to pay a record $1.6 billion penalty. Most of the penalty amounts to forced spending on improving pipeline safety. A section of PG&E pipeline exploded in San Bruno in 2010, killing eight people. The company says it is working to become the safest energy company in the U.S.

Table 1: Example summaries that are human-written (abstractive), and produced by extractive and our systems. Colored text indicates important details not present in the human-written summary.

1 Introduction

Language model pre-training has advanced the state-of-the-art in many NLP tasks ranging from sentiment analysis, question answering, natural language inference, named entity recognition, and textual similarity; more recently, they have been used in summarization (Liu and Lapata, 2019; Lewis et al., 2020). State-of-the-art pre-trained models include GPT (Radford et al., 2018), BERT (Devlin et al., 2019), BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020).

These models acquire prior syntactic and semantic knowledge from large text corpora and are further fine-tuned on task-specific smaller datasets, such as news article-summary pairs. However, specifics of test set news articles might not be well represented in the training set. For example, a news publisher might train a summarizer on dated news and wants to summarize latest incoming news. This suggests potential improvements if the summarizer learns these specifics before summaries are generated. In this work, we explore *transductive learning* (Vapnik, 1998) by adapting a fine-tuned summarizer to the test set by learning from its input articles.

The main obstacle for *transduction* is the absence of a reliable training signal, as no references are available in test time. Therefore, we propose constructing extractive references by selecting summarizing sentences from the input text by a separately trained model. Summarizing sentences are often fused and compressed to form abstractive summaries (Lebanoff et al., 2019), and contain additional important details providing better context, as illustrated in Table 1. Further, we use a denoising objective to predict summarizing sentences conditioned on masked input articles. In this way, the model balances the copying and generation dynamic (See et al., 2017; Gehrmann et al., 2018; Bražinskas et al., 2020) as not all information for accurate summary predictions is available in the masked input. To further preserve summary abstractivness, we predict a small portion of abstractive summaries (∼5% on CNN/DM) from the annotated training set. This results in only a small fraction of the training time needed to perform transduction ($<$ 4% on CNN/DM[1]). Moreover, we leverage summarizing sentences from training set inputs in the fine-tuning phase by predicting both abstractive and extractive references. As we show, this method outperforms standard fine-tuning on abstractive references alone. Finally, we show improvements in the scenario when only dated news articles with summaries are available for training and the aim is to summarize recent news articles in test time.

All in all, we empirically demonstrate that our model (TRSUM), that utilizes summarizing sentences in the fine-tuning and transduction phases, significantly improves the quality of summaries. Besides achieving state-of-the-art results on standard datasets (CNN/DM (Hermann et al., 2015) and NYT (Sandhaus, 2008)), it also yields more coherent and abstractive summaries. Our main contributions can be summarized as follows:

- we present the first application of transductive learning to summarization;
- we show state-of-the-art results on standard summarization datasets (CNN/DM and NYT);
- we show that transduction is beneficial for summarizing more recent CNN news [2].

[1]On an AWS 8-GPU p3.8xlarge instance, full training took 9 hours while transduction only 15 minutes.

[2]The codebase will be publicly available.

2 Joint Fine-Tuning

Our model (TRSUM) has a Transformer encoder-decoder architecture (Vaswani et al., 2017), which is initialized with pre-trained BART (Lewis et al., 2020). Before we learn from the test set articles using transductive learning (presented in Sec. 3), we jointly fine-tune the model on extractive and abstractive references in the training set. Extractive references are useful for learning, as they often contain omitted details in abstractive summaries and provide additional context to the reader, see Table 1.

Let $\{x_i, y_i\}_{i=1}^{N}$ be article-summary pairs in the training set. First, we greedily select k sentences from the input article x that maximize the ROUGE score[3] to the summary y by following Liu and Lapata (2019). We concatenate these sentences to form an extractive summary $\hat{y}$ that is word-by-word predicted using teacher-forcing (Williams and Zipser, 1989). Further, to prevent trivial solutions, we randomly mask words in x with a special mask token[4]. Intuitively, this forces the decoder to balance between copying from the input and generating novel content (See et al., 2017; Gehrmann et al., 2018; Bražinskas et al., 2020). Finally, we formulate a *joint fine-tuning* objective in Eq. 1. We also illustrate the whole procedure in Fig. 1.

$$\frac{1}{N}\sum_{i=1}^{N}\log p_\theta(y_i|x_i) + \frac{1}{M}\sum_{j=1}^{M}\log p_\theta(\hat{y}_j|\hat{x}_j) \quad (1)$$

Notice that the joint objective in Eq. 1 re-uses the model's architecture without a specialized task embedding. The model can easily differentiate between abstractive and extractive summary prediction/generation as only in the latter the input contains a special mask token. We validate this in an ablation experiment presented in Sec. 6.2.

Lastly, our main goal is to learn an abstractive summarizer $p_\theta(y|x)$ without overfitting on extractive references. Thus, we control for the ratio of abstractive and extractive instances N and M, respectively, by drawing decisions from the Bernoulli distribution $Bern(\alpha)$. If α is set to 0, it results in abstractive pairs only.

[3]We used the average of ROUGE-1 and ROUGE-2 F scores.

[4]We also experimented with masking only summarizing sentences. However, this lead to inferior results.

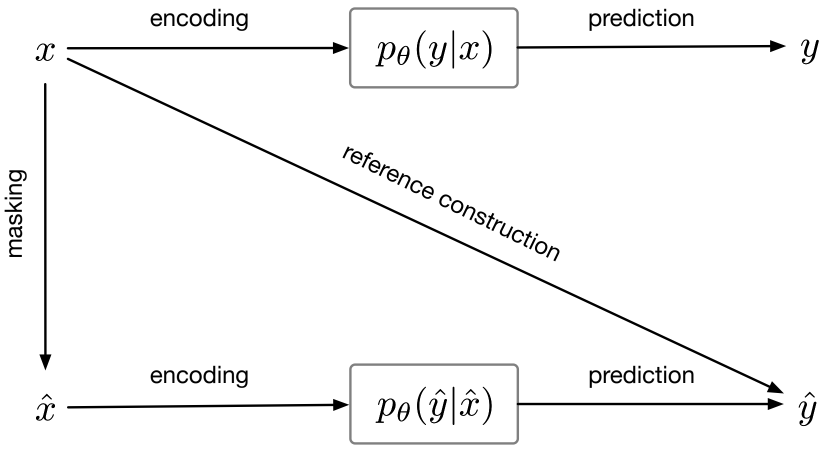


Figure 1: Illustration of the joint objective and the associated procedure. Here we randomly mask the input article x resulting in $\hat{x}$. Further, construct $\hat{y}$ by concatenating summarizing sentences in x. Lastly, we jointly predict abstractive and extractive references y and $\hat{y}$, respectively.

3 Transduction

Consider a scenario where a news publishing agency has a fine-tuned model on dated article-summary pairs and wants to summarize upcoming news articles for which summaries are not yet available. In this setting, an immediate response might not be necessary and latency can be traded for summary quality. In this light, we propose to leverage *transductive learning* (Vapnik, 1998) and further fine-tune the model by learning from test set input articles. First, we train an extractive summarizer that predicts summarizing sentences, as explained in Sec. 3.1. Second, we extract summarizing sentences from test set input articles and construct references $\hat{y}$. Lastly, we optimize the model by predicting these references using $p_\theta(\hat{y}|\hat{x})$ in Eq. 1.

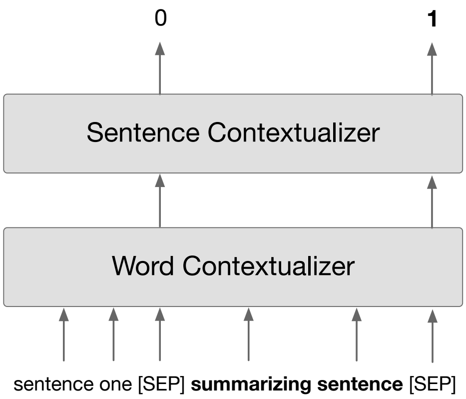


Figure 2: Extractive summarizer contextualizes words and subsequently sentences. The final outputs are binary tags where 1 indicates a summarizing sentence.

3.1 Extractive Summarizer

To produce extractive references on the test set, we train an extractive summarizer. The summarizer consists of two Transformer encoders and predicts which sentences are summarizing, as illustrated in Fig. 2. Formally, let $[s_1, s_2, ..., s_m]$ denote sentences in an article where each sentence is separated by a special symbol ([SEP]). Further, let $[b_1, b_2, ..., b_m]$ be their associated binary tags where 1 indicates a summarizing sentence.

To compute model predictions for sentences, we proceed as follows. First, we feed the sequence of concatenated sentences $[s_1, s_2, ..., s_m]$ to the first encoder and obtain sentence representations $[e_1, e_2, ..., e_m]$. Intuitively, these representations capture semantics of each sentence useful for determining their salience and how well they summarize the whole article. To better capture cross-sentence dependencies, we feed the sentence representations to the second encoder and obtain contextualized representations $[c_1, c_2, ..., c_m]$. Finally, we feed each representation c_i to a feed-forward neural network $f_\theta(c_i)$ to obtain scores.

3.2 Regularization

In transduction, when the model is solely optimized for predicting extractive summaries, the previously learned abstractive summarization and its performance can degradate (Goodfellow et al., 2013; Kemker et al., 2017); see Sec. 6.2 for a confirming experiment. As a form of regularization, we propose to additionally predict abstractive summaries from the training set using the full objective in Eq. 1. In practice, we found that sampling a similar amount of training pairs as in the test set (about

Year	Count	Avg. # words	Avg. # sents
2016	12799	34.65	2.46
2017	11292	32.49	2.34

Table 2: CNN summary statistics for more recent years.

5% on CNN/DM) to be sufficient. Another point of consideration is that the extractive summarizer, presented in Sec. 3.1, can erroneously select non-summarizing sentences, resulting in less reliable references. Consequently, we found it beneficial to also add extractive pairs from the training set created using a heuristic presented in Sec. 2.

3.3 Tracking of Overfitting

Tracking of overfitting is essential for model development. To monitor overfitting during transduction, we propose the following simple procedure. First, we sample a tiny subset of validation pairs (around 1,000). To closely resemble transduction, we produce extractive references using the extractive model presented in Sec. 3.1. Further, we combine the validation extractive pairs with the training and test set pairs used for transduction (see Sec. 3.2). Finally, we track ROUGE-L scores on the validation human-written abstractive references. This, in turn, allows us to determine when abstractive summarization performance starts to decrease to perform early stopping.

4 Experimental Setup

4.1 Datasets

The evaluation was performed on two main summarization datasets: CNN/DailyMail (Hermann et al., 2015) and New York Times (NYT) (Sandhaus, 2008). CNN/DM contains news articles and associated highlights, i.e., a few bullet points giving a brief overview of the article. We used the standard splits of 287k, 13k, and 11k for training, validation, and testing, respectively. We did not anonymize entities and followed See et al. (2017) to pre-process the first sentences of CNN. For NYT, we used a provided dataset used in (Liu and Lapata, 2019), which consists of 38264, 4002, 3421 training, validation, and test set instances, respectively. The instances are news articles accompanied by short human-written summaries, where summaries shorter than 50 words were removed.

The original CNN/DM dataset contains news from 2007 to 2015. To test whether transduction is beneficial for more recent news, we obtained newer snapshots of CNN, namely for 2016 and 2017. We downloaded CNN articles published in 2016 and 2017 using NewsPlease,[5] extracted raw contents, and retained those having a story highlight as a summary in the beginning of the article. The statistics are shown in Table 2. These sets were used for transduction only.

Finally, we truncated input documents to 1000 subwords [6] by preserving complete sentences. To monitor overfitting, we used 1k, 500, and 100 validation instances for transduction on CNN/DM, NYT, and CNN 2016/2017, respectively. In all experiments, we used ROUGE-L for the stopping criterion. For evaluation, we used the standard ROUGE package (Lin, 2004) and report F1 scores.

4.2 Human Evaluation

For human evaluation experiments, we randomly sampled 300 articles from CNN/DM test set. Further, we generated and compared summaries from BART + FT and TRSUM. We used Amazon Mechanical Turk (AMT) and ensured that only high-quality workers could participate. We asked workers to pass a custom qualification test, which only 14.6% of those who took it passed. For further details, see Appendix 11.1. Finally, we requested 3 annotators per HIT and used MACE (Hovy et al., 2013) to estimate annotator competences and recover the most likely answer per HIT accordingly.

4.3 Model Details

For pre-initialization, we used the large pre-trained BART model (Lewis et al., 2020) available with FairSEQ. We also used a subword tokenizer with maximum of 50k subwords. The model had 12 layers both in the encoder and decoder and a hidden size of 1024. In total, it consisted of 400M parameters. During fine-tuning and transduction, the architecture remained unchanged.

During joint fine-tuning (TRSUM$^-$), presented in Sec. 2, we masked 25% of words in input articles, and set $\alpha = 0.1$ to produce on average 10% of extractive instances at each epoch. In transduction, we masked 10% of input words, and sampled 14k and 5k training instances at each epoch for CNN/DM and NYT, respectively. Here, α was set to 0.1. In all experiments, we used Adam (Kingma and Ba, 2014) for weight updates, and beam search for summary generation with 3-gram

[5] https://github.com/fhamborg/news-please

[6] The maximum number of subwords includes sentence separator special tokens.

blocking (Paulus et al., 2017). All experiments were performed on 8-GPU p3.8xlarge Amazon instance. For CNN/DM, we performed joint fine-tuning for 6 epochs and transduction for 3 epochs. For NYT, 9 epochs of joint fine-tuning and 3 epochs for transduction.

4.4 Extractive Summarizer

To obtain extractive references for transduction (EXTREF), we used the BART's fine-tuned encoder, and an additional transformer encoder (Vaswani et al., 2017) to contextualize sentence representations. For CNN/DM, we set the number of layers to 3, and attention heads to 16. For NYT, we set it to 2 layers with attention heads number to 8. To produce binary scores, we used a linear transformation that is followed by the sigmoid function.

To select summarizing sentences from the training set input articles, we used a greedy heuristic (ORACLE) that maximizes ROUGE scores between the summarizing sentences and the gold summary as in Nallapati et al. (2016a); Liu and Lapata (2019). We selected up to 3 sentences per input article. In inference, we ranked candidate sentences by scores and selected top-3 sentences. Also, we applied N-gram blocking during selection to avoid repetitive content as in Liu and Lapata (2019). Given a current extractive summary s and candidate sentence c, we skip c if there exists a trigram overlap between c and s.

5 Evaluation Results

5.1 Automatic Evaluation

Standard Datasets We report automatic evaluation based on ROUGE F1 on the CNN/DM and NYT test sets, the results are shown in Table 3.

First of all, we observed that joint fine-tuning (TRSUM^{-}), which utilizes both extractive and abstractive summaries of the training set, outperforms the standard fine-tuning that utilizes only the former. Second, we observed that transduction further improves the performance of the jointly fine-tuned model on both datasets. We also performed an independent-samples t-test to compare our full model to BART+FT. It indicates that all results are statistically significant under $p < 0.05$ except ROUGE-2 on NYT.

Recent News It is common to assume training and test sets to share a common distribution (Quadrianto et al., 2009; Kann and Schütze, 2018). However, in practice, this assumption might need to be violated (Ueffing et al., 2007). For instance, we might want to transduct a summarizer on fresh news while it was fine-tuned on more dated news. To test our approach, we used a summarizer jointly fine-tuned on the standard CNN/DM training set, spanning news from 2007 to 2015 and transducted on more recent CNN news (2016 and 2017). The results are presented in Table 4.

First of all, we observed that joint fine-tuning is superior to the standard one on both datasets. Second, even though extractive noisy references (EXTREF) have low ROUGE scores, we further improve the results by performing transduction. Moreover, when higher quality extractive references were used, namely produced using the oracle heuristic (TRSUM \w ORACLE), additional improvements were observed. This shows that our approach is beneficial for settings where training set and test set distributions are different.

5.2 Human Evaluation

To gain deeper insights into how extractive references affect the coherence of summaries our model generates, we performed a human evaluation study. Additionally, we evaluated the factual consistency of generated summaries, which is an open problem in summarization (Kryscinski et al., 2020; Maynez et al., 2020).

Coherence In evaluation, generated summaries were presented in a random order, as well as the input article and reference summary for context. For each HIT, we asked the 3 annotators which of the two generated summaries, if any, was more coherent. We gave the following definition: *"The more coherent summary has better structure and flow, is easier to follow. The facts are presented in more logical order."* The TRSUM model was preferred 110 times (22.0%), while BART + FT was preferred 89 times (17.8%). In 101 cases (20.2%), the annotators indicated that none of the two summaries was preferable. We conclude that the TRSUM summaries were significantly more coherent than the BART + FT summaries ($p < 0.05$ using a one-sided z-test).

We observed that CNN/DM articles tend to be more coherent than the associated bullet point summaries. Further, we observed that summarizing sentences we used for learning (EXTREF) tend to be among lead 5 (61.1%) with a very small gap between them (0.529 sentences on average). There-

	CNN/DailyMail			New York Times		
	R1	R2	RL	R1	R2	RL
ORACLE	55.21	32.86	51.36	61.70	42.23	58.34
LEAD-3	40.42	17.62	36.67	38.28	19.75	34.96
Extractive / Compressive						
SUMMARUNNER (Nallapati et al., 2016a)	39.60	16.20	35.30	-	-	-
REFRESH (Narayan et al., 2018)	40.00	18.20	36.60	-	-	-
SUMO (Liu et al., 2019)	41.00	18.40	37.20	42.30	22.70	38.60
COMPRESS (Durrett et al., 2016)	-	-	-	42.20	24.90	-
JETS (Xu and Durrett, 2019)	41.70	18.50	37.90	-	-	-
BERTSUMEXT (Liu and Lapata, 2019)	43.25	20.24	39.63	46.66	26.35	42.62
MATCHSUM (Zhong et al., 2020)	44.41	20.86	40.55	-	-	-
Abstractive						
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	43.71	26.40	-
BOTTOMUP (Gehrmann et al., 2018)	41.22	18.68	38.34	-	-	-
DRM (Paulus et al., 2017)	-	-	-	42.94	26.02	-
BERTSUMEXTABS (Liu and Lapata, 2019)	42.13	19.60	39.18	49.02	31.02	45.55
PEGASUS (Zhang et al., 2020)	44.17	21.47	41.11	-	-	-
BART + FT (reported) (Lewis et al., 2020)	44.16	21.28	40.90	-	-	-
BART + FT (ours)[7]	44.01	21.13	40.81	52.97	35.19	49.32
Ours						
TRSUM$^-$	44.59	21.58	41.50	53.55	35.54	49.81
TRSUM	**44.96**	**21.89**	**41.86**	**53.72**	**35.72**	**50.06**
EXTREF	43.93	21.12	40.20	47.49	27.57	43.88

Table 3: ROUGE F1 scores on the standard CNN/DM and New York Times test sets.

	CNN 2016			CNN 2017		
	R1	R2	RL	R1	R2	RL
ORACLE	53.05	36.87	49.89	52.58	36.97	49.57
LEAD-3	31.87	16.62	29.06	28.82	14.32	26.21
BERTSUMEXTABS	33.17	14.43	30.56	30.44	12.51	27.98
BART + FT	34.93	15.83	32.14	32.62	14.27	29.98
TRSUM$^-$	35.40	15.92	32.62	32.92	14.21	30.24
TRSUM	**35.58**	**16.32**	**32.78**	**33.07**	**14.63**	**30.45**
TRSUM \w ORACLE	36.10	16.72	33.27	33.37	15.01	30.71
EXTREF	32.14	15.37	29.17	29.19	13.30	26.43

Table 4: ROUGE F1 scores on more recent CNN test sets. In TRSUM \w ORACLE we used ORACLE extractive references transduction.

fore, we hypothesize that the model learns from consecutive sentences more natural text structures that emanate in summaries.

Factual Consistency For evaluating factual consistency, each HIT presented one input article and one generated summary from BART + FT or TRSUM. To simplify the task, we focused the workers' attention on a single highlighted sentence per summary, which we picked at random, and asked if that sentence, as shown in the context of the full summary, is factually consistent with the article. We gave detailed guidelines and examples for factual errors, see Appendix 11.1. Effectively, this setup measured how likely a randomly chosen summary sentence is factually consistent with the summarized article. We found that 263 of the 300 BART + FT summary sentences (87.7%) were judged factual, compared to 254 for the 300 TRSUM summaries (84.7%). This is a small difference that we

	R1	R2	RL
BART + FT	44.01	21.13	40.81
BART + FT + TR	44.83	21.79	41.69
TrSum$^-$	44.59	21.58	41.50
TrSum	**44.96**	**21.89**	**41.86**

Table 5: Comparison between transduction of the BART model that was fine-tuned using our and the default method on the CNN/DM test set.

	N1	N2	N3
Gold	0.178	0.528	0.718
BART + FT	0.019	0.101	0.186
BART + FT + TR	0.026	0.132	0.234
TrSum$^-$	0.028	0.135	0.238
TrSum	**0.029**	**0.145**	**0.254**

Table 6: The proportion of novel n-grams on the standard CNN/DM test set.

found not statistically significant ($p < 0.05$ using a one-sided z-test).

6 Analysis

6.1 Transduction of Fine-tuned BART

We further explored whether it is possible to perform transduction of the BART model that was already fine-tuned only on abstractive summaries (BART + FT). The results on the standard CNN/DM dataset are presented in Table 5. They indicate that transduction is beneficial and noticeably improves the results. We hypothesize that the model also benefits from the training set extractive instances that are predicted. However, it does not reach the results achieved by our full approach.

6.2 Ablation

To gain insights into the inner workings of transduction, we performed an ablation study by removing components from models fine-tuned jointly and only on abstractive references. We plot the ROUGE-L scores on the validation subset that was used for transduction in Fig. 3.

First of all, we observed that masking is important in both cases, and without it the models degradate. We believe that the mask token is used as a mode indicator for the decoder. And without it, the decoder is unable to differentiate the two modes (extractive vs abstractive summary prediction). Second, we observed that the removal of the training set instances, as explained in Sec. 3.2, makes TrSum$^-$ converge to the same ROUGE score as the extractive references used for transduction. On the other hand, it makes BART + FT degradate. Finally, without ablations, we observed two different learning dynamics. BART + FT initially decreases in the ROUGE score for 2 epochs, and then slowly starts to improve by surpassing the baseline extractive references at epoch 4. We hypothesize that it is caused by unfamiliarity with predicting extractive summaries. On the other hand, TrSum experiences only a minor decrease in the beginning, possibly due to the lower quality of extractive references of the test set tagged by a model in Sec. 3.1, and then it steadily improves.

[7]We used shorter input with only complete sentences that we believe resulted in a slightly worse performance.

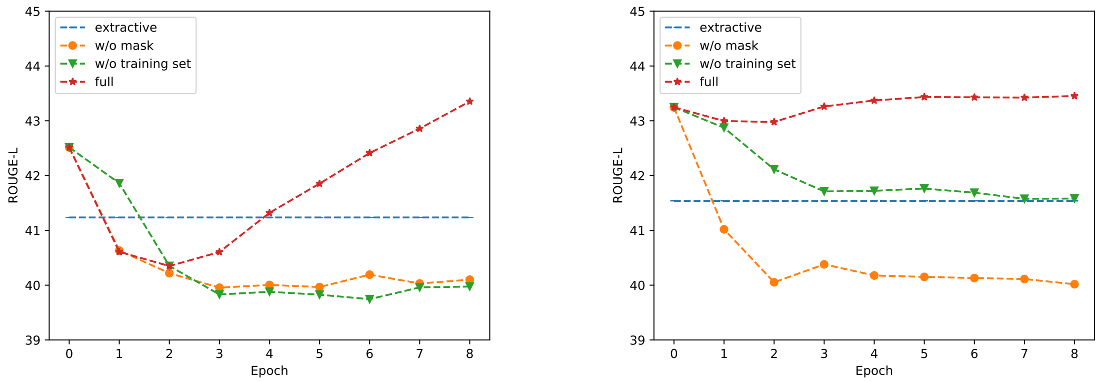
6.3 Novel N-grams

We also analyzed generated summaries in terms of the proportion of novel n-grams that appear in the produced summaries but not in the source texts. The results are shown in Table 6. We observed that joint fine-tuning and transduction increase the proportion of novel n-grams, thus making summaries more abstractive. By comparing extractive and abstractive summaries, we noticed the selected sentences in extractive summaries often paraphrase sentences in the abstractive ones. We hypothesize that the exposure to the references with paraphrases allows the model to generate more variant summaries.

7 Related Work

Single-document extractive and abstractive summarization is a well-established field with a large body of prior research (Dasgupta et al., 2013; Rush et al., 2015; Nallapati et al., 2016b; Tan et al., 2017; See et al., 2017; Fabbri et al., 2020; Laban et al., 2020).

The utilization of extractive summaries to improve abstractive summarization has also received some recent attention. Commonly, in a two-step procedure where summarizing fragments are first selected, and then paraphrased into abstractive summaries (Chen and Bansal, 2018; Bae et al., 2019). Alternatively, to alter attention weights (Hsu et al., 2018; Gehrmann et al., 2018) to bias the model to rely more on summarizing input content. Finally, to perform pre-training on extractive references prior to abstractive summarization (Liu and Lapata, 2019). In our case, we predict extractive references word-by-word by constructing a denoising objec-



(a) BART + FT (b) TRSUM

Figure 3: Ablation during the transduction phase. ROUGE-L scores on the 1k subset of the standard CNN/DM validation set; *extractive* indicates the extractive references used for transduction.

tive that also masks input words. We use the same model without modifications, and predict extractive and abstractive references jointly.

Transductive learning has been applied to a number of language-related tasks, such as machine translation (Ueffing et al., 2007), paradigm completion (Kann and Schütze, 2018), syntactic and semantic analysis (Ouchi et al., 2019), and more recently to style transfer (Xiao et al., 2021). However, to the best of our knowledge, transductive learning has never been applied to summarization.

More recently, PEGASUS (Zhang et al., 2020) leveraged text fragments for pre-training. The text fragments are selected using heuristics, such as top-K sentences. Instead, we utilize a separate extractive model or gold summaries to select sentences that form extractive references.

8 Conclusions

In this work, we present the first application of *transductive learning* to summarization. We propose learning from summarizing sentences extracted from the test set's input articles to better capture their specifics. We additionally propose a mechanism to regularize and validate the transductive model. The proposed method achieves state-of-the-art results in automatic evaluation on the CNN/DM and NYT datasets, and it generates more abstractive and coherent summaries. Finally, we demonstrate that transduction is useful when trained on dated news and transducted on more recent news.

9 Future Work

First, learning from single data points in the online fashion can be a promising direction. This, in turn, could call for the decoder's modularization that is less prone to overfitting. This could be achieved using more efficient fine-tuning methods, such as adapters (Houlsby et al., 2019) and continuous prefixes (Li and Liang, 2021). Second, we believe that content fidelity can be improved by learning from the test set's input using specialized methods. Third, where training and test sets are in different domains, adaptation in the transduction phase can be fruitful, similar to Ueffing et al. (2007).

10 Ethics Statement

Human Evaluation We used a publicly available service (Amazon Mechanical Turk) to hire voluntary participants, requesting native speakers of English. The participants were compensated above the minimum hourly wage in their self-identified countries of residence.

Dataset The dataset was collected and used in accordance to non-commercial personal purpose permitted by the data provider.

References

Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sang-goo Lee. 2019. Summary level training of sentence rewriting for abstractive summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. pages 169–174.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of Association for Computational Linguistics (ACL)*.

Anirban Dasgupta, Ravi Kumar, and Sujith Ravi. 2013. Summarization through submodularity and dispersion. In *Proceedings of Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of Association for Computational Linguistics (ACL)*.

Alexander R Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2020. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. *arXiv preprint arXiv:2010.12836*.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H. Hovy. 2013. Learning Whom to Trust with MACE. In *Proc. of NAACL*.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of Association for Computational Linguistics (ACL)*.

Katharina Kann and Hinrich Schütze. 2018. Neural transductive learning and beyond: Morphological generation in the minimal-resource setting. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. 2017. Measuring catastrophic forgetting in neural networks. *arXiv preprint arXiv:1708.02072*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Philippe Laban, Andrew Hsi, John Canny, and Marti A Hearst. 2020. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5135–5150.

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Scoring sentence singletons and pairs for abstractive summarization. *arXiv preprint arXiv:1906.00077*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of Association for Computational Linguistics (ACL)*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the*

11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582–4597, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries acl. In *Proceedings of Workshop on Text Summarization Branches Out Post Conference Workshop of ACL*, pages 2017–05.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yang Liu, Ivan Titov, and Mirella Lapata. 2019. Single document summarization as tree induction. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of Association for Computational Linguistics (ACL)*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016a. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016b. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The Conference on Computational Natural Language Learning (SIGNLL)*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of Association for Computational Linguistics (ACL)*.

Hiroki Ouchi, Jun Suzuki, and Kentaro Inui. 2019. Transductive learning of neural language models for syntactic and semantic analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Novi Quadrianto, James Petterson, and Alex Smola. 2009. Distribution matching for transduction. *Advances in Neural Information Processing Systems*, 22:1500–1508.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of Association for Computational Linguistics (ACL)*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of Association for Computational Linguistics (ACL)*.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of Association for Computational Linguistics (ACL)*.

Vladimir Vapnik. 1998. Statistical learning theory wiley. *New York*, 1.

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

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Fei Xiao, Liang Pang, Yanyan Lan, Yan Wang, Huawei Shen, and Xueqi Cheng. 2021. Transductive learning for unsupervised text style transfer. *arXiv preprint arXiv:2109.07812*.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. 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 3292–3303, Hong Kong, China. Association for Computational Linguistics.

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

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

11 Appendices

11.1 Details on the Mechanical Turk Setup

Custom Qualification Test. For all our evaluations on Mechanical Turk, we first created a custom qualification test that could be taken by any worker from a country whose main language is English, who has completed 100 or more HITs so far with an acceptance rate of 95% or higher. The qualification test consisted of three questions from our factual consistency setup; two of which had to be answered correctly, along with an explanation text (5 words or more) to explain when ”not factually consistent” was chosen. 53% of workers who started the test provided answers to all three questions, and 27.6% of these answered at least two correctly and provided a reasonable explanation text, i.e., only 14.6% of the test takers were granted the qualification. The qualification enabled workers to work on our factual consistency HITs as well as our HITs judging summary coherence.

Payment and Instructions. The coherence task took workers a median time of 125 seconds per HIT, for which we paid \$0.40 with a bonus pf \$0.20, amounting to an hourly rate of \$17. The factual consistency task took workers a median time of 30 seconds per summary; the payment was \$0.12 plus a bonus of \$0.05, amounting to an hourly rate of \$20. This task was relatively quick to do as a single summary sentence had to be judged; we also highlighted article sentences that are semantically similar to the highlighted summary sentence, in order to make the relevant information from the article more quickly accessible for fact checking.[8] The factual consistency task contained instructions shown in Fig. 4. The instructions for the coherence task are quoted in the main text above.

Excluding Spammers. For both tasks, we ran code attempting to automatically detect potential spammers and label them for exclusion, in order to ensure high quality annotations. Anyone labeled for exclusion was disqualified for further HITs, their HIT answers were excluded from the results and HITs were extended to seek replacement answers. For the coherence task, any worker who spent less than 10 seconds per HIT was labeled for exclusion. For the factual consistency task, the minimum time per HIT required was 5 seconds; in addition; workers who wrote very short explanation texts for their ”not factually consistent” answers (median length 3 words or less) were excluded. We also added 10 HITs with known factuality, and workers who answered 3 or more of them but with an accuracy less than 2/3 were excluded as well. Any worker who was not excluded according to the above criteria received the bonus.

[8] We used the cosine distance of the universal sentence embeddings (Cer et al., 2018) to measure semantic similarity.

Please evaluate whether the blue sentence from the summary is consistent with the information in the article.
Select **no** if the blue sentence is not consistent, i.e., its facts are not supported by the article.

Select **no** in cases like these:

- The blue sentence **contradicts** information in the article. The blue sentence might say "A fire broke out in Seattle", but the article says it broke out in Portland. Or the blue sentence might say "the Republicans won the election", but the article indicates that the Democrats won instead.
- The blue sentence **adds** a fact that is not mentioned anywhere in the article. For example, the blue sentence might say that "A fire broke out at 2am", but the article doesn't mention the time when the fire broke out.

Figure 4: Instructions for evaluating if a summary sentence (highlighted in blue) was factually consistent with the source article.

Inter-Annotator Agreement and MACE For the binary factual consistency evaluation, 521 of the 600 HITs (86.8%) had a full agreement of all 3 workers; all other HITs had two agreements. For the coherence evaluation, in which 3 different answers were possible (first or second summary more coherent; or none), 258 of the 300 HITs (86.0%) had an agreement of 2 or more workers per HIT. As noted in the main text above, we ran MACE (Hovy et al., 2013) to further improve upon these raw answers by unsupervised estimation of worker trustworthiness and subsequent recovery of the most likely final answer per HIT.