Curriculum Data Augmentation for Low-Resource Slides Summarization

Anonymous ACL submission

Abstract

Data augmentation is commonly used in training in low-resource scenarios. However, there are sometimes large discrepancy between distributions of augmented data and target data. How to bridge the gap between the augmented and target data, especially when target data is harder-to-learn? In this paper, we study improved data augmentation strategies in the scenario of scientific slides text summarization, where we generate a textual summary based on texts of presentation slides. Since slides are messy and difficult to understand by current models, we introduce an easier form of data, i.e., articles in natural language. The basic idea is that we generate the transition data between slides and articles, and all three of them form a curriculum for neural models to learn the distribution transition from article data to slides data. We find that our approach achieves consistent improvements over different backbone summarization models. The curriculum-oriented data augmentation method can generate data that fill the gap between the easy-to-obtain data and the low-resource task data. We show that curriculum learning and data augmentation can be combined to help NLP models learn from otherwise hard-to-learn data. [1]

1 Introduction

Nowadays, presentation slides have become one of the main materials for disseminating ideas and thoughts in conferences, lectures or events. Similar to other formats of documents such as articles, presentation slides grow rapidly in volume. The difficulty for readers to exhaustively go through all contents for so many slides calls for the automatic summarization of such documents. With textual summaries for presentation slides, readers can easily find ones they are interested in and then dive into details.

Most existing works focus on summarizing articles for large-scale datasets, e.g., CNN/Daily Mail (Hermann et al., 2015), XSum (Narayan et al., 2018) and PubMed/ArXiv (Cohan et al., 2018), leaving the summarization of slides an unsolved puzzle. Traditional extractive text summarization methods (Mihalcea and Tarau, 2004; Zhang et al., 2018; Liu, 2019). do not trivially apply to noisy and broken text as slides, as seen in Figure 1. On the other hand, abstractive methods are developed to generate summaries for given text (Rush et al., 2015; See et al., 2017; Gehrmann et al., 2018). In recent years, summarization with large-scale pre-trained language models (PLMs) (Devlin et al., 2019; Raffel et al., 2020; Lewis et al., 2020; Zhang et al., 2020) has become the dominant paradigm, due to their excellent ability in language modeling. However, slides differ greatly from regular text these models are pre-trained or fine-tuned on, leaving the potential of the PLMs not fully realized.

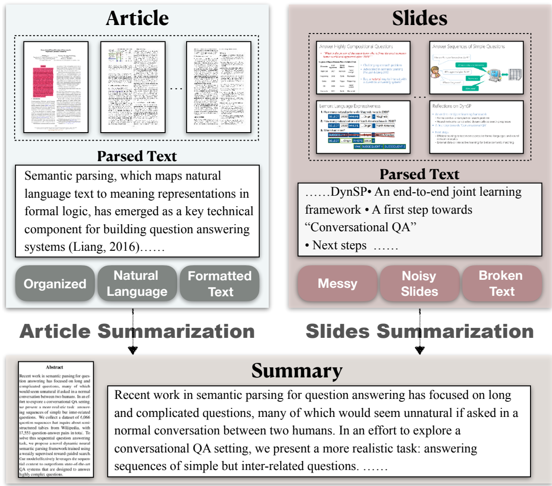


Figure 1: Difference between slides summarization and the corresponding article summarization.

To fill in the blank, we propose the task of slides summarization in this work. There are two major challenges in this task: *limited resource* and *noisy input*. Summarizing slides is a low-resource task by nature, because we can not utilize any section

[1] We will release the code after paper's acceptance.

of the slides as summaries directly for building large-scale datasets. In contrast, article summarization often uses the first paragraph or abstract of an article as ground-truth summary. Moreover, the rich multi-modal information (i.e., textual, visual and layout information (Xu et al., 2020b, 2021)) in the source documents (i.e., presentation slides) are difficult to be utilized for current summarization techniques. When parsed into text, the source documents are much more noisy and ill-formatted than articles, posing great challenges for language models to understand in a low-resource setting.

Instead of focusing on designing summarization models, we tackle this task from the perspective of data augmentation. The intuition is that language models can learn better representations of unfamiliar slides if they start with something familiar, i.e., articles in natural language. We are motivated by the idea of *curriculum learning* (CL) (Bengio et al., 2009), where models benefit from starting small and gradually increasing the learning difficulty when training with limited data samples (Wang et al., 2021; Wu et al., 2020). In this sense, the accompanying articles facilitate the learning of the model by providing connections between what it knows well (natural language) and what it does not (noisy slides).

In this paper, we propose the LESSON framework for **L**ow-r**E**source **S**lides **S**ummarizati**ON** with curriculum data augmentation. Different from traditional CL (Bengio et al., 2009; Liu et al., 2018; Platanios et al., 2019), we do not discover curriculum data within training data. Instead, we build extra curriculum data from the accompanying articles of slides. To bridge the gap between data distributions of articles and slides, we enrich the curriculum by generating in-between data samples, yielding a sequence of data samples to language models in an increasing order of difficulty. We adopt a simple generative model to realize the transition from articles to slides, which is controlled by a balancing weight in training objectives. Like CL, our method focuses on optimizing training and thus does not need such corresponding articles for slides in the test time. In this sense, LESSON is practical, because we do not usually have corresponding articles for test-time slides.

In summary, the contributions of this paper include: *1)* We are the first to propose the slides summarization problem, and tackle low-resource scientific slides summarization as a concrete task; *2)* We propose LESSON framework for this task, a training strategy with curriculum data augmentation, in order to learn a robust model for slides summarization; *3)* We conduct extensive experiments to verify and understand the effectiveness of the LESSON framework.

2 Related Work

Data Augmentation In NLP, the goals of data augmentation include increasing training data size (Fadaee et al., 2017), achieving regularizing effects (Hernández-García and König, 2020; Fabbri et al., 2021), and diversifying data (Lu et al., 2020; Kumar et al., 2019). Common data augmentation approaches include rule-based methods, interpolation methods and model-based methods (Feng et al., 2021). Rule-based approaches utilize a set of pre-defined rules to transform existing data and create new samples (Wei and Zou, 2019). Interpolation approaches construct a continuous hidden space for text and interpolate new data from that hidden space (Chen et al., 2020; Cheng et al., 2020). Model-based approaches either leverage a pre-trained language model (Anaby-Tavor et al., 2020; Yang et al., 2020) or train an auxiliary model based on existing training data (Kumar et al., 2019), and then use the model to generate new samples.

Curriculum Learning Curriculum learning (Bengio et al., 2009) is a learning strategy that mimics the human learning process. Theoretically, curriculum can be regarded as an optimization strategy (Bengio et al., 2009) for non-convex training criteria. Starting from easier samples, the model can learn a smoother training objective, and approximate local minima that have better generalization ability towards the global minima (Wang et al., 2021). Previous empirical results have demonstrated the strengths of curriculum learning in a wide range of NLP tasks, such as question answering (Liu et al., 2018), natural language understanding (Xu et al., 2020a), machine translation (Platanios et al., 2019), and text classification (Wei et al., 2021). These works have testified to curriculum learning's ability to learn from noisier data (Wu et al., 2020), reduce training time (Wu et al., 2020; Platanios et al., 2019), and gain performance improvements over randomized training (Liu et al., 2018; Platanios et al., 2019; Xu et al., 2020a).

Curriculum Data Augmentation Previously, Wei et al. (2021) explore curriculum data augmentation in a few-shot setting. They use rule-based approaches (Wei and Zou, 2019) to augment the original dataset through controlled noising, i.e., creating noisier and more difficult data for curriculum learning. In contrast, we employ curriculum learning to make the optimization process smoother and improve the model's performance on the hard data. Both of our studies represent a new approach in curriculum learning: instead of discovering curriculum from existing data, we artificially create data of different difficulty levels.

Abstractive Summarization The goal of abstractive summarization is to generate concise and precise summary text for documents. Traditionally, such methods mostly applies sequence-to-sequence encoder-decoder architectures (Rush et al., 2015; See et al., 2017; Gehrmann et al., 2018; You et al., 2019). In recent years, pre-trained transformer-based (Vaswani et al., 2017) language models have achieves remarkable success on a wide range of NLP tasks (Radford et al., 2019; Devlin et al., 2019; Liu et al., 2019b; Raffel et al., 2020; Lewis et al., 2020). Many approaches using pre-trained language models have achieved state-of-the-art results on abstractive summarization tasks (Liu and Lapata, 2019; Rothe et al., 2020; Zhang et al., 2020; Beltagy et al., 2020). Most summarization studies work on text summarization, and some focus on summarizing from noisier input such as meeting transcripts and dialogues (Liu et al., 2019a; Zhao et al., 2019; Liu et al., 2019c; Zhu et al., 2020). In this work, we directly use parsed text as the input due to the difficulty in slides structure parsing. Thus, our input is even noisier with structured information mixed in plain text. As our proposed LESSON is model-agnostic, we leave the handling of slides structures and layouts to future work.

3 Proposed Approach

In this section, we detail the slides summarization task and the proposed LESSON method, a curriculum-based data augmentation training framework for slides summarization.

3.1 Task Formulation

Let $\mathcal{X}^S$ denote slides text and $\mathcal{Y}$ denote summary text, and we have $\mathbb{D}^S = (\mathcal{X}^S, \mathcal{Y})$ for a slides summarization dataset. We hypothesize that slides data are harder to learn compared with normal article data, given limited parsing techniques and language models that are easier to process natural language.

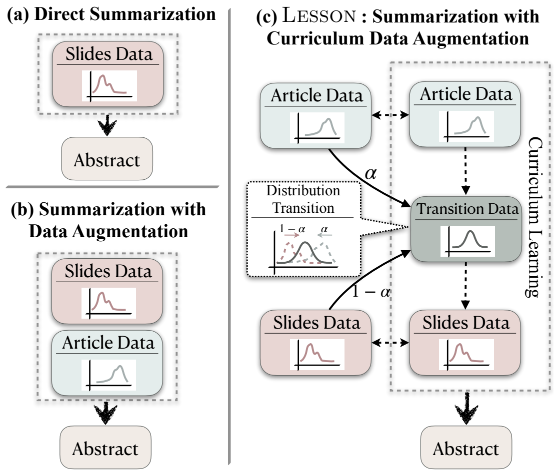


Figure 2: Comparison of (a) Direct summarization, input $\mathcal{X}^S$ slides, output $\mathcal{Y}$ summary and (b) Data augmentation with $\mathcal{X}^A$ article and (c) LESSON. The acquisition of transition data $\mathcal{X}^{A\to S}$ is shown.

Inspired by previous studies on curriculum learning, which demonstrate that learning from easier samples helps a model to learn harder samples, we introduce paralleled article data $\mathcal{X}^A$ as augmentation, denoted as $\mathbb{D}^A = (\mathcal{X}^A, \mathcal{Y})$. Thus we have paralleled data triples $(\mathcal{X}^S, \mathcal{X}^A, \mathcal{Y})$. Since the distribution of the parallel articles is greatly different from target slides data, we further generate transition data to bridge the distributional discrepancy between them, which is denoted as $\mathcal{X}^{A\to S}$ and thus $\mathbb{D}^{A\to S} = (\mathcal{X}^{A\to S}, \mathcal{Y})$.

Finally, we set up a multi-stage ($\mathcal{X}^A$, $\mathcal{X}^{A\to S}$, $\mathcal{X}^S$) curriculum learning strategy to learn from easier samples to harder ones during training. Hence, the model is able to deal with slides data directly during inference.

3.2 Transition Data Generation

We approximate the transition between slides data $\mathcal{X}^S$ and article data $\mathcal{X}^A$, in the form of text generation, where we train a transition model $P_{A\to S}$ by fine-tuning a BART (Lewis et al., 2020), a pre-trained sequence-to-sequence model.

In order to control the transition process, we design a hyper-parameter schedule weight α in the training objective $\mathcal{L}_{A\to S}$, which is defined as

$$\mathcal{L}_{A\to S} = \alpha\mathcal{L}_S + (1-\alpha)\mathcal{L}_A \quad (1)$$

where $\mathcal{L}_S$ is the cross entropy loss for article-to-slides generation, and $\mathcal{L}_A$ is the cross entropy loss for article-to-article generation, which is basically input reconstruction. Note that a larger α empirically steers the generated text towards the distribution of slides, and vice versa.

After training the article-to-slides model, we use it for generating intermediate transition data $\mathcal{X}^{A\rightarrow S} = P_{A\rightarrow S}(\mathcal{X}^A, \alpha)$. The generated data is then compiled to form a transition data pair $\mathbb{D}^{A\rightarrow S} = (\mathcal{X}^{A\rightarrow S}, \mathcal{Y})$, reaching the full augmented dataset with quadruples: $\mathbb{D} = (\mathcal{X}^S, \mathcal{X}^A, \mathcal{X}^{A\rightarrow S}, \mathcal{Y})$.

3.3 Curriculum Learning

Finally, we employ curriculum learning strategy to train models on the augmented dataset $\mathbb{D}$.

Curriculum learning strategies usually include two components: the difficulty measurer and the curriculum scheduler. The difficulty measurer evaluates the difficulty level of the training samples. In this work, we employ a pre-defined approach to difficulty measurement that slides data are harder to learn compared with article data in natural text. Specifically, we treat $\mathbb{D}^A$, $\mathbb{D}^{A\rightarrow S}$, and $\mathbb{D}^S$ as discrete training buckets that are trained during different curriculum stages. Note that the curriculum is fully extendable, as we can generate more stages of transition data $\mathbb{D}^{A\rightarrow S}$ by tuning the schedule weight α.

During each of the stages, the model takes as input $\mathcal{X}^\star$ and is optimized with a simple text summarization objective for summaries $\mathcal{Y}$. To automatically regulate curriculum stages, we observe the development set ROUGE score during training and move to the next curriculum stage when the score converges. In this way, the model starts with the easiest article data, then switch to the transition data, and finally learns the target slides data. During inference, the model is enhanced with the knowledge to understand noisy data and does not require the help of parallel article data, which may be hard to collect during test time.

4 Experimental Setup

4.1 Datasets

We use two slides summarization datasets for experiments, where one is an existing dataset with few adaptations and the other is collected ourselves. Statistics of these datasets are reported in Table 1, where they are randomly split into training, development, and test set at the ratio of 7:1:2. One *major difference* between them is the *noisiness* of the parsed slides text, due to different slides parsing techniques. Empirically, noisier slides text as input makes it harder for models to generate summaries. We describe the details of these datasets as follows.

Dataset	Split	# Inst.	Src. Len.	Tgt. Len.
S^3	Train	191	697	138
	Dev	26	817	145
	Test	55	663	132
NoisyS^3	Train	306	2,878	298
	Dev	44	2,388	202
	Test	87	2,444	319

Table 1: Statistics of the datasets. "# Inst." denotes the number of instances in the splits. "Src. Len." and "Tgt. Len." denote the *average* token number of parsed slides text (source) and summary text (target) respectively.

The Existing Dataset: S^3 We use the public SciDuet-ACL dataset released by Sun et al. (2021), which is a dataset for generating presentation slides from scientific papers.[2] Based on SciDuet-ACL, we construct the dataset of slides summarization by keeping the abstract of a paper as the summary for the corresponding slides, referred to as S^3 (**S**cientific **S**lides **S**ummarization). We keep only the slides text parsed by Sun et al. (2021) and discard the figures and tables in the slides, which are identified by OpenCV (Bradski, 2000).

The Collected Dataset: NoisyS^3 We also collect a relatively larger-scale dataset from scratch. We crawl from the ACL anthology and scholars' home pages for pairs of slides and scientific papers and get 137 and 300 instances respectively. Since most of the slides are in PDF format, we transform all slides into PDF and use *pymupdf* to parse slides text.[3] In other words, we do not exclude figures and tables, thus the parsed text is much noisier than one in S^3 due to mixed numbers and text snippets.

4.2 Metrics & Baselines

Since the output of our task is a resemblance to the mainstream text summarization tasks, we evaluate the generated summaries with the commonly-used ROUGE-1/2/L (Lin, 2004), which calculates the n-gram overlap of generated and reference text.

There was no previous work for slides summarization, so we adopt several widely used

[2] The authors of SciDuet only release the ACL portion of the dataset due to copyright issues.

[3] https://github.com/pymupdf/PyMuPDF

Model	The S^3 Dataset			The NoisyS^3 Dataset		
	R-1	R-2	R-L	R-1	R-2	R-L
Transformer	3.50	0.01	3.42	2.00	0.00	2.00
BERT2BERT	22.87	4.55	13.59	21.44	3.61	12.13
+ Lesson	24.97(+9.18%)	4.55(+0.00%)	13.79(+1.47%)	24.14(+12.59%)	4.07(+12.74%)	12.80(+5.52%)
SciB2SciB	24.89	4.10	12.94	21.22	3.50	11.56
+ Lesson	26.62(+6.95%)	4.97(+21.22%)	12.91(-0.23%)	23.78(+12.06%)	4.25(+21.43%)	12.14(+5.02%)
BART	36.65	9.46	20.25	30.05	6.72	16.68
+ Lesson	38.55(+5.18%)	10.30(+8.88%)	21.19(+4.64%)	34.21(+13.84%)	8.48(+26.19%)	18.27(+9.53%)
BART-ArXiv	37.46	9.64	20.77	32.18	7.46	17.19
+ Lesson	38.92(+3.90%)	10.30(+6.85%)	21.56(+3.80%)	35.88(+11.50%)	9.61(+28.82%)	19.02(+10.65%)

Table 2: Main summarization results of baselines with Lesson in two datasets. Lesson shows consistent performance improvements over baseline models.

abstractive text summarization models as baselines, where we prioritize on the Transformer-based generative language models pre-trained on large corpus.[4] These models follow the dominant encoder-decoder architecture for sequence-to-sequence (Seq2Seq) generation. Note that Lesson is open to the choices of the backbone summarization model.

Transformer (Vaswani et al., 2017) is the classic attention-based Seq2Seq model, which serves as a non-pre-trained baseline for this task.

BERT2BERT (Rothe et al., 2020) leverages pre-trained checkpoints, such as BERT (Devlin et al., 2019), to initialize both encoder and decoder, where the only variables initialized randomly is the encoder-decoder attention. In addition, we also use SciBERT (Beltagy et al., 2019) as the pre-trained checkpoint for in-domain scientific text.

BART (base version) (Lewis et al., 2020) is a transformer-based PLM for Seq2Seq generation, pre-trained with a set of self-supervised sequence denoising tasks. BART has shown its effectiveness on a variety of text generation tasks, which makes it our primary backbone model for Lesson in the experiments.

BART-ArXiv We also study the effect of *transfer learning* from article summarization to slides summarization, since these two tasks are in a similar domain but the former is much more high-resourced than the latter. To this end, we finetune a BART (base version) with article summarization objective on ArXiv dataset (Cohan et al., 2018), which consists of 215,913 scientific papers.[5]

[4]Note that we do not incorporate some summarization models such as T5 (Raffel et al., 2020) and Pegasus (Zhang et al., 2020) as baselines because part of the summary data in S^3 is leaked in the C4 dataset upon which these models are pre-trained.

4.3 Implementation Details

We use pre-trained checkpoints provided by HuggingFace (Wolf et al., 2019) in our experiments.

For transition data $\mathcal{X}^{A\rightarrow S}$ generation, we fine-tune a BART model for 3 epochs on the training set. We set the maximum generation length to 1024, the maximum length for BART. We use Adam (Kingma and Ba, 2015) as the optimizer for all of our models and set the learning rate to 5×10^{-5}.

For scheduling Lesson training, we train our models for 20 epochs in each curriculum stage and select the checkpoint with the highest development set ROUGE score to enter the next stage.

Each model has a different maximum input token length configuration. Since our input data is usually longer than most of the models' maximum possible input length, we set each of them to the max possible length. The max input token lengths for BERT- and BART- models are set to 512, 1024. For summary generation, we set the minimum length to 50 tokens and maximum to 400 tokens. We also set the number of beams to 4 and length penalty to 2.0.

5 Results & Analysis

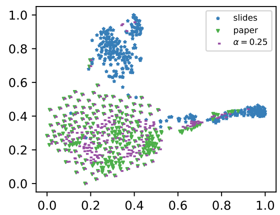
5.1 Main Results

We report the main results of slides summarization in Table 2. In general, we observe a consistent performance boost with Lesson across different base models, including out-of-the-box pre-trained language models (BERT2BERT, BART), models pre-trained on in-domain texts (SciB2SciB), and

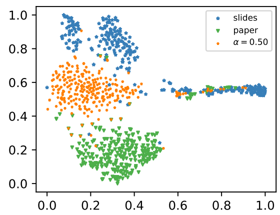
[5]We ensure there is no data leakage in the ArXiv dataset.

Dataset	Ablation	R-1	R-2	R-L
S^3	BART	36.65	9.46	20.25
	+ $\mathcal{X}^A$	36.84	9.37	19.98
	+ $\mathcal{X}^A$, $\mathcal{X}^{A\to S}$	35.88	8.24	19.13
	+ $\mathcal{X}^A$ + CL	37.25	9.65	20.73
	+ LESSON	**38.55**	**10.30**	**21.19**
NOISYS^3	BART	30.05	6.72	16.68
	+ $\mathcal{X}^A$	32.83	7.25	17.35
	+ $\mathcal{X}^A$, $\mathcal{X}^{A\to S}$	30.28	6.65	16.41
	+ $\mathcal{X}^A$ + CL	33.93	7.42	17.53
	+ LESSON	**34.21**	**8.48**	**18.27**

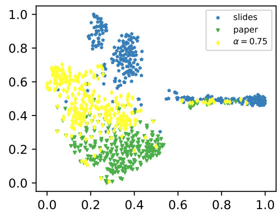
Table 3: Ablation study on different components of LESSON using BART as the base model, including article data $\mathcal{X}^A$, transition data $\mathcal{X}^{A\to S}$, and curriculum learning strategy CL. If "+ CL" is not indicated, the data is trained in random order. "+ LESSON" is equivalent to "+ $\mathcal{X}^A$, $\mathcal{X}^{A\to S}$ + CL". A leap in performance emerges when data augmentation and curriculum learning are combined.



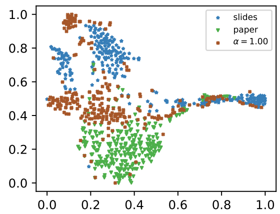
(a) Visualization of $\mathcal{X}^A$, $\mathcal{X}^{A\to S}_{\alpha=0.25}$, and $\mathcal{X}^S$).



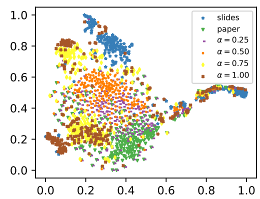
(b) Visualization of $\mathcal{X}^A$, $\mathcal{X}^{A\to S}_{\alpha=0.50}$, and $\mathcal{X}^S$).



(c) Visualization of $\mathcal{X}^A$, $\mathcal{X}^{A\to S}_{\alpha=0.75}$, and $\mathcal{X}^S$).



(d) Visualization of $\mathcal{X}^A$, $\mathcal{X}^{A\to S}_{\alpha=1.00}$, and $\mathcal{X}^S$).



(e) Visualization of all.

Figure 3: t-SNE visualization of the SciBERT embeddings of articles, transition data and slides text in order to show the distribution shift within curriculum. t-SNE is computed individually for each sub-figure. Therefore, the same data distribution in different sub-figures are slightly different. The transition data is controlled by the schedule weight α. Best viewed in color.

pre-trained models finetuned on downstream in-domain summarization task (BART-ArXiv). The ROUGE-2 boosts are the greatest in most cases, up to 28.82% for BART-ArXiv.

When we compare LESSON-BART with BART-ArXiv, we find that a base model with LESSON outperforms the base model with transfer learning on much larger in-domain data. It shows that in cases where abundant in-domain texts are not available, LESSON can still achieve similar performance under the low-resource constraint.

We also observe bigger performance improvements on the NOISYS^3 dataset. This corroborates LESSON's ability to learn noisy and difficult data better through curriculum learning.

Ablation Study We perform an ablation study on different components of LESSON with BART, the results are presented in Table 3. We observe that there is little effect when we use data augmentation without curriculum learning ("+$\mathcal{X}^A$" and "+ $\mathcal{X}^A$, $\mathcal{X}^{A\to S}$ "). However, if we use the curriculum to strategically order the training of the augmented data, we can take full advantage of $\mathcal{X}^A$ and $\mathcal{X}^{A\to S}$. This provides strong evidence for our assumption that combining data augmentation and curriculum learning leads to better performance on hard-to-learn data.

5.2 Analysis

In this section, we analyze the curriculum data augmentation in LESSON to figure out two research questions: Does the augmented data form a curriculum? Does curriculum learning work for slides summarization? [6]

Data Visualization Theoretically, CL helps the model learn from the easy distribution (articles) to the hard distribution (slides). For an empirical understanding of the distribution shift between them, we visualize the SciBERT embedding (Beltagy et al., 2019) of articles, slides, and in-between transition data in the curriculum as seen in Figure 3. The ideal transition data bridges the distribution gap between the slides data and the article data. Our qualitative evaluation concludes that Figure 3(b) ($\mathcal{X}^{A\to S}_{\alpha=0.5}$) shows the most desirable case, where the generated transition data lies in the middle of the article and slides data. For other configurations

[6] For the rest of the experiments, we run LESSON with BART (base) by default.

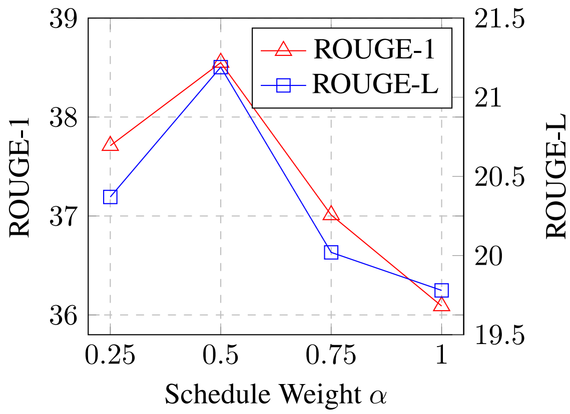


Figure 4: Results of LESSON-BART using transition data generated from different schedule weight α. The result indicates that $\mathcal{X}^{A\to S}_{\alpha=0.5}$ yields the best performance, which is consistent with our qualitative evaluation that $\mathcal{X}^{A\to S}_{\alpha=0.5}$ makes up the smoothest transition between the article data and the slides data.

of α, the transition data does not make as good a distributional shift between the article and the slides data.

Then, we verify how $\mathcal{X}^{A\to S}$ generated with different α affects the curriculum. The results are presented in Figure 4, which we find is consistent with the pattern we find in Figure 3. The transition data $\mathcal{X}^{A\to S}_{\alpha=0.5}$ achieve the best result because it forms the smoothest distributional shift between the article data $\mathcal{X}^{A}$ and the slides data $\mathcal{Y}$. Other transition data, especially $\mathcal{X}^{A\to S}_{\alpha=1.0}$, performs poorly because the transitions are not smooth, which would lead to a harder optimization process.

Multi-Staged Curriculum We extend the curriculum to multiple stages. In particular, we include $\{\mathcal{X}^{A\to S}_{\alpha=i}\}_i$ as $\mathcal{X}^{A\to S}$. The results are shown in Table 4. We organize stage order based on Equation 1. Hence, $\mathcal{X}^{A\to S}_{\alpha=0.25}$ is closest to the article data and $\mathcal{X}^{A\to S}_{\alpha=1.00}$ is closest to the slides data.

On the S^3 dataset, the single-stage curriculum learning strategy proves to be the best, and we observe performance decrease when we add more stages to training. This is because, as shown previously in Figure 3, the transition data for $\alpha \in \{0.25, 0.75, 1.00\}$ do not make up a smooth distributional shift between the article data and the target slides data. The learning process would be complicated by these noisy transition data.

Curriculum Schedules To study the effects of curriculum learning, we schedule the curriculum learning in three settings: 1) *regular curriculum* (CL) as described in Section 3.3; 2) *anti-curriculum* (Anti-CL) where we reverse the training order of the curriculum; and 3) *random curriculum* (Rand-CL) where we shuffle all data during training. As seen in Table 5, we observe LESSON trained with regular CL achieves the best result whereas LESSON with random CL has the lowest scores, which is consistent with the findings of Wu et al. (2020). The results also demonstrate the difficulty order of slides, transition data, and articles, which is consistent with our intuition and findings mentioned above.

$\{\mathcal{X}^{A\to S}_{\alpha=i}\}_i$ Setting	R-1	R-2	R-L
$i = 0.50$ (LESSON baseline)	**38.55**	**10.30**	**21.19**
$i \in \{0.25, 0.50\}$	35.76	6.87	18.01
$i \in \{0.25, 0.50, 0.75\}$	31.73	5.00	14.95
$i \in \{0.25, 0.50, 0.75, 1.00\}$	29.94	3.99	15.06

Table 4: Results of LESSON-BART with multiple curriculum stages for different values of α on S^3. Increasing curriculum stages leads to performance drop, as the transition data for $\alpha \in \{0.25, 0.75, 1.00\}$ do not make up a smooth distributional shift between the article data and the target slides data.

Dataset	Schedule	R-1	R-2	R-L
S^3	CL	**38.55**	**10.30**	**21.19**
	Anti-CL	36.06	10.03	19.58
	Rand-CL	35.88	8.24	19.13
NOISYS^3	CL	**34.21**	**8.48**	**18.27**
	Anti-CL	32.03	7.16	17.23
	Rand-CL	30.28	6.65	16.41

Table 5: Results of LESSON-BART trained with different curriculum schedules. Anti-CL refers to the inverse training order (first $\mathcal{X}^S$, then $\mathcal{X}^{A\to S}$, and last $\mathcal{X}^A$). Curriculum learning shows better results than Anti-CL and random training schedule.

5.3 Case Study

We examine the generation results from a baseline BART model and BART equipped with LESSON on the NOISYS^3 test set and compare them with the reference abstract and the input slides. We can identify some key observations in the results, as demonstrated in Table 6. Due to the difficulty of the input slides, the BART baseline model is unable to generate meaningful summarization. On the other hand, as we expected, the improvement of LESSON over the baseline is particularly visible. LESSON makes it possible to digest the noisy input and provide more consistent and factually correct

Reference
Conformity is a type of social influence involving a change in opinion or behavior to fit in with a group. Employing several social networks as the source for our experimental data, we study how the effect of conformity plays a role in changing users' online behavior. We formally define several major types of conformity in individual, peer, and group levels. We propose Confluence model to formalize the effects of social conformity into a probabilistic model. Confluence can distinguish and quantify the effects of the different types of conformities. To scale up to large social networks, we propose a distributed learning method that can construct the Confluence model efficiently with near-linear speedup. Our experimental results on four different types of large social networks, i.e., Flickr, Gowalla, Weibo and Co-Author, verify the existence of the conformity phenomena. Leveraging the conformity information, Confluence can accurately predict actions of users. Our experiments show that Confluence significantly improves the prediction accuracy by up to 5-10% compared with several alternative methods.
BART Baseline
Aspect Term Extraction (ASE) is a novel way to measure the sentiment of an aspect term. In this paper, we propose a new dataset, the Aspect-based Sentiment Estimation (AGE) dataset. Aspect term extraction (aspect-term-extraction) is used to quantify the frequency of distinct aspect terms. We propose a novel approach, which is based on a dataset that is used for the task of estimating the frequency and frequency of the most frequent aspect terms in a domain. Our dataset, which consists of two datasets, consists of three datasets, and consists of four datasets. The first two datasets (the second dataset) are based on the same dataset, and the second dataset on the third dataset (the third dataset). The second dataset, our dataset consists of a dataset consisting of two different datasets, which are both based on our dataset. The third dataset is a dataset based on two datasets. We evaluate our dataset on a variety of domain-specific domains, and we show that our dataset outperforms the previous datasets on a wide range of domains. We show that we outperform the previous dataset on both domains. We also show that aspect term Extraction is more accurate than the first dataset. In our dataset, we find that our approach outperforms our previous dataset by more than 90% of the time. We use a new approach to evaluate the accuracy of our datasets.
LESSON-BART
Aspect-based sentiment analysis estimates the sentiment expressed for each particular aspect (e.g., battery, screen). Different words or phrases, however, may be used to refer to the same aspect, and similar aspects may need to be aggregated at coarser or finer granularity to fit the available space or satisfy user preferences. We introduce the problem of aspect similarity at multiple levels of learning. We decompose it in two processing phases, to allow previous work on term similarity and hierarchical clustering to be reused. We show that the second phase, where aspects are clustered, is almost a solved problem. We also introduce a novel sense pruning mechanism for WordNet-based similarity measures, which improves their performance in the first phase. Finally, we provide publicly available benchmark datasets.

Consistent with the slides. Inconsistent with the slides or flawed generation.

Table 6: Case study for generated slides summaries for different methods on the NOISYS^3 test set.

summarization. There are, however, several problems in LESSON's generation that we observed: *1)* LESSON sometimes generates contents that are consistent with the input slides, but inconsistent within the passage. The example in Table 6 shows LESSON can point out that three factors are influencing social conformity, but it is unable to name them, making the marked sentence abrupt in the summary; *2)* LESSON sometimes generates factually inconsistent information. For example, the datasets mentioned in Table 6's LESSON generation is incorrect. These errors show that as a generation model in its essence, LESSON still faces common obstacles in natural language generation.

6 Conclusion and Future Work

In this paper, we tackle the slides summarization problem, which is under-studied but of much practical use. We formulate this problem as a text summarization task, and propose LESSON with curriculum data augmentation to overcome the *limited resource* and *noisy input* challenges in this task. Experiments on both the public dataset S^3 and our collected dataset NOISYS^3 show that LESSON consistently improves summarization results over baseline models. Further analyses show the data augmentation process successfully creates transition data that bridges the gap between the article data and the slides data. The transition data enables curriculum training, which proves to boost the model's ability to learn from the noisy slides data. In the future, we will emphasize on multi-modal slides summarization to utilize the layout information of the slides, and explore few-shot adaptation to slides of unseen domains.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7383–7390. AAAI Press.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. AdvAug: Robust adversarial augmentation for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5961–5970, Online. Association for Computational Linguistics.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexander Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2021. Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 704–717, Online. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, T. Mitamura, and E. Hovy. 2021. A survey of data augmentation approaches for nlp. *ArXiv*, abs/2105.03075.

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

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.

Alex Hernández-García and Peter König. 2020. Data augmentation instead of explicit regularization.

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*.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, Volume 1 (Long and Short Papers), pages 3609–3619, Minneapolis, Minnesota. Association for Computational 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. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Curriculum learning for natural answer generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4223–4229. ijcai.org.

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019a. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1957–1965. ACM.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhengyuan Liu, Angela Ng, Sheldon Lee, Ai Ti Aw, and Nancy F. Chen. 2019c. Topic-aware pointer-generator networks for summarizing spoken conversations.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. *Gender Bias in Neural Natural Language Processing*, pages 189–202. Springer International Publishing, Cham.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy X. R. Wang. 2021. D2S: Document-to-slide generation via query-based text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1405–1418, Online. Association for Computational Linguistics.

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.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jason Wei, Chengyu Huang, Soroush Vosoughi, Yu Cheng, and Shiqi Xu. 2021. Few-shot text classification with triplet networks, data augmentation, and curriculum learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5493–5500, Online. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

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. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2020. When do curricula work? *arXiv preprint arXiv:2012.03107*.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020a. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. 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 2579–2591, Online. Association for Computational Linguistics.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020b. Layoutlm: Pre-training of text and layout for document image understanding. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Yongjian You, Weijia Jia, Tianyi Liu, and Wenmian Yang. 2019. Improving abstractive document summarization with salient information modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2132–2141, Florence, Italy. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784, Brussels, Belgium. Association for Computational Linguistics.

Zhou Zhao, Haojie Pan, Changjie Fan, Yan Liu, Linlin Li, and Min Yang. 2019. Abstractive meeting summarization via hierarchical adaptive segmental network learning. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 3455–3461. ACM.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.