Curricular Next Conversation Prediction Pretraining for Transcript Segmentation

Anvesh Rao Vijjini[1*] **Hanieh Deilamsalehy**[2]
Franck Dernoncourt[2] **Snigdha Chaturvedi**[1]
[1]UNC Chapel Hill
{anvesh,snigdha}@cs.unc.edu
[2]Adobe Research
{deilamsa,franck.dernoncourt}@adobe.com

Abstract

Transcript segmentation is the task of dividing a single continuous transcript into multiple segments. While document segmentation is a popular task, transcript segmentation has significant challenges due to the relatively noisy and sporadic nature of data. We propose pretraining strategies to address these challenges. The strategies are based on "Next Conversation Prediction" (NCP) with the underlying idea of pretraining a model to identify consecutive conversations. We further introduce "Advanced NCP" to make the pretraining task more relevant to the downstream task of segmentation break prediction while being significantly easier. Finally we introduce a curriculum to Advanced NCP (Curricular NCP) based on the similarity between pretraining and downstream task samples. Curricular NCP applied to a state-of-the-art model for text segmentation outperforms prior results. We also show that our pretraining strategies make the model robust to speech recognition errors commonly found in automatically generated transcripts.

1 Introduction

Text segmentation is the task of identifying segment breaks to organize a continuous text into semantically independent segments. Prior research in text segmentation has largely focused on segmenting documents such as Wikipedia articles (document segmentation) (Lukasik et al., 2020; Zhang et al., 2019; Badjatiya et al., 2018; Koshorek et al., 2018) or dialogues such as chat or text messages (dialogue segmentation) (Hsueh et al., 2006; Arguello and Rosé, 2006; Xia et al., 2022; Xing and Carenini, 2021). In this paper we address text segmentation of transcripts (transcript segmentation). Figure 1 shows examples of segments in transcript data. Transcript segmentation can help summarize long videos, podcasts or meetings by segmenting and summarizing the transcript such as "Video chapters" in YouTube videos or "Outline Generation" (Zhang et al., 2019) from documents.

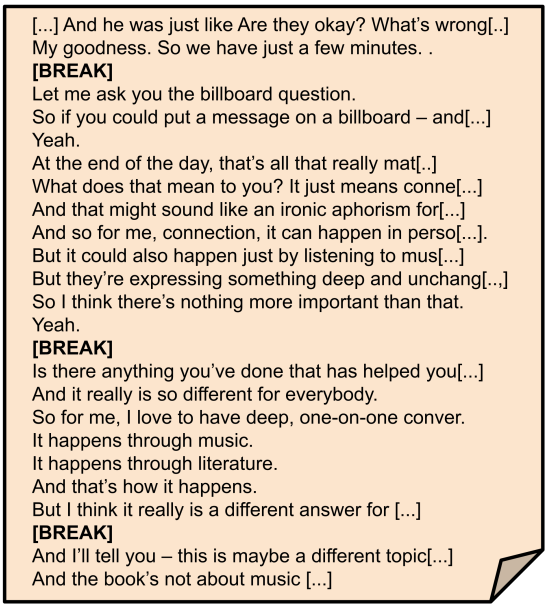

Figure 1: Transcript Segmentation example from the SliceCast-Podcast (Midei and Mandic, 2019) dataset. Here each line indicates start of a new sentence and segment breaks are noted with "[BREAK]".

However, only few works have addressed segmentation of transcripts (Midei and Mandic, 2019; Jing et al., 2021; Gruenstein et al., 2008). As shown in Figure 1, transcripts consist of a mix of short sentences, utterances, interjections and long form document style answers. Unlike Wikipedia articles or chat, the sporadic and non uniform flow of text in transcripts makes annotation of segment breaks hard even for humans (Gruenstein et al., 2008). Furthermore, transcripts often involve Automatic Speech Recognition errors such as insertions, deletions, replacement and lack of proper punctuation which add to the challenges. As a result of these challenges, most labeled transcript segmentation datasets are small in size making it difficult for models to be trained on them.

To address this issue, we propose pretraining strategies that can be useful in resource constrained

* Work done during internship at Adobe Research.

Findings of the Association for Computational Linguistics: EACL 2023, pages 2597–2607

settings where huge labeled datasets are not available. Our first strategy is Next Conversation Prediction (NCP). In this strategy, pairs of conversations are classified into 1 or 0 based on whether they contiguously in the transcript or not. We hypothesize that the effectiveness of this pretraining task relies on its similarity and relevance to the segmentation task. Our second strategy, Advanced Next Conversation Prediction (Advanced NCP), introduces conditions on the NCP pretraining data to increase the relevance of the pretraining strategy to the segmentation task. Our third strategy is Curricular NCP where we pretrain the model in two distinct phases based on which pretraining samples are closest to the task of transcript segmentation.

Our experiments show that the application of the proposed pretraining strategies on multiple segmentation architectures outperforms their corresponding non pretrained versions. Also, NCP does not rely on segmentation labels. We show that it is a strong unsupervised approach that outperforms state-of-the-art unsupervised model for transcript segmentation. Finally, we observe that the pretraining strategies makes the model more robust to noise and better at predicting highly segmented regions of a transcript.

Our contributions are:

- Propose a pretraining strategy based on Next Conversation Prediction for transcript segmentation. We show that it also acts as a strong unsupervised approach for this task.
- Propose Advanced NCP and Curricular NCP pretraining strategies based on similarity and relevance of pretraining samples to segmentation task.
- Provide a new state-of-the-art in transcript segmentation.
- Evaluate robustness of proposed pretraining strategies to noisy training data.
- We perform additional analysis to investigate the errors made by the pretrained models.

2 Related Work

Text segmentation has been addressed in both unsupervised (Solbiati et al., 2021; Glavaš et al., 2016) and supervised manner (Midei and Mandic, 2019; Lukasik et al., 2020; Koshorek et al., 2018; Badjatiya et al., 2018) with early works focusing on unsupervised techniques (Hearst, 1997; Choi, 2000; Utiyama and Isahara, 2001; Eisenstein, 2009). However, since the definition of a segment, could be highly domain and data dependant, supervised learning is desirable.

Koshorek and Cohen (2017) and Koshorek et al. (2018) use LSTMs to identify if a sentence ends a segment or not. Similarly, Li et al. (2018) use GRUs and pointer-generator networks for this task. These works in segmentation propose a hierarchical approach, where the sentences are encoded into a fixed size representation followed by mapping the representations to a sequence of binary labels whether the current segment is ending at this sentence or not. Badjatiya et al. (2018) use attention based CNN-LSTMs and phrase the task differently by providing inputs of a median sentence and its right and left context to identify segment breaks. Lukasik et al. (2020) simplify the new paradigm for this task by using the left and right contexts with respect to an end of a sentence as input. They were also the first to use large pretrained language models for this task. They establish a new SOTA in text segmentation. Hence, we use their model as the base model in our pretraining experiments.

Very few works have focused on transcript segmentation. Midei and Mandic (2019) provide a podcast dataset for research in this domain and propose an LSTM and Universal Sentence Encoder (Cer et al., 2018) based sequence labeling model. Jing et al. (2021) identify introductions in podcast transcripts using BERT (Devlin et al., 2019). Solbiati et al. (2021) propose an unsupervised technique for meeting transcript segmentation. They use large language model representations including Sentence BERT (Reimers and Gurevych, 2019) and BERT to compute cosine similarity between subsequent conversations and estimate segment breaks. We present a new pretraining strategy aimed towards addressing transcript segmentation but not specific to any one model architecture. Our work is also related to Curriculum Learning (Bengio et al., 2009). It has gained popularity among NLP tasks such as Sentiment Analysis (Cirik et al., 2016) Question Answering (Sachan and Xing, 2016), NLG (Liu et al., 2018) and the GLUE benchmark (Xu et al., 2020). More recently, some works have used curriculum learning in the pretraining process of large language models. Wang et al. (2020) propose curriculum learning for pretraining the encoder of their speech translation system on multiple speech based tasks of varying difficulty. Nagatsuka et al. (2021) gradually introduce longer samples to BERT's pretraining to observe performance improvements in

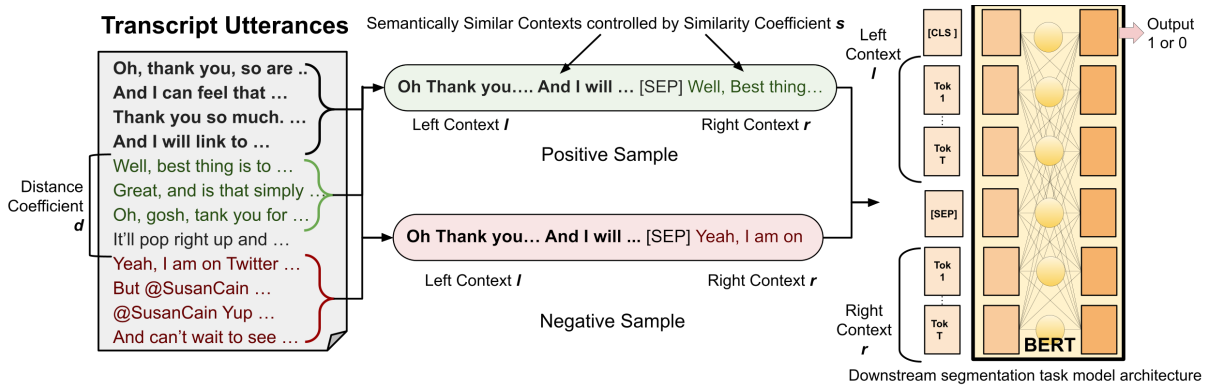

Figure 2: Advanced Next Conversation Prediction (NCP) pretraining strategy illustrated with roles of the coefficients in controlling pretraining task difficulty and similarity to the downstream task.

resource poor settings. In our curriculum learning setting, we order pretraining samples based on their similarity to the downstream segmentation task.

3 Text segmentation as a binary classification task

There are two major ways supervised text segmentation has been addressed in the past. First, by phrasing it as a sequence labeling problem, where each sentence has a label indicating if it ends the segment or not (Midei and Mandic, 2019; Lukasik et al., 2020; Koshorek et al., 2018). As a result an entire transcript forms a single training instance consisting of a sequence of binary labels.

Second, by phrasing text segmentation as a binary classification task where we provide left context and right context around a sentence end and predict if the two contexts belong to the same segment or not. In particular, the input in this task is two text segments - left context (l) and right context (r) of the end of a sentence, each T tokens in length. The output in this task is 0 - if l and r belong to two different segments (segment break) and 1 if l and r belong to the same segment (not a segment break). In this setting, the number of instances is proportional to the number of segments. Lukasik et al. (2020) note in their experiments that the second setting outperforms the first on transcript segmentation. We follow the second setting and refer to it as the segmentation task in the rest of this paper.

4 Curricular Next Conversation Prediction

In this section, we explain the proposed pretraining strategy. First, we explain our basic pretraining strategy - Next Conversation Prediction (NCP). Then, we present improvements on this strategy to make the pretraining task easier and more relevant to the transcript segmentation task (Advanced NCP). Finally, we introduce curriculum learning to our pretraining strategy (Curricular NCP) that presents the pretraining instances in an order that is more helpful for the segmentation task.

4.1 Next Conversation Prediction

Next Conversation Prediction is the backbone of our proposed pretraining strategies. Large language models such as BERT have gained recent success on the segmentation task (Lukasik et al., 2020). One of the pretraining tasks in BERT is NSP (Next Sentence Prediction). In NSP, a sentence pair is provided as input and the model predicts if the sentences occurred consecutively in their original corpora. We hypothesize that BERT's success in text segmentation might be attributed to the NSP pretraining's similarity to the segmentation task.

Motivated by NSP, we propose a pretraining strategy based on Next Conversation Prediction (NCP) for the task of transcript segmentation. In NCP, we address a binary classification task. The input is a pair of transcript contexts and the output is a label indicating whether the contexts are adjacent or not. Specifically, the input consists of the left (l) and right (r) contexts of a sentence end (T tokens each). The output is 1 if the two contexts are adjacent and 0 otherwise. Note that NCP does not use any information about segment break labels and so can potentially be used on transcripts dataset without segment break annotations.

NCP has two major advantages over NSP. First, NCP has longer contexts making the model learn information from a wide range of sentences varying in content and style. Second, NCP as pretraining task makes the pretraining step similar to the seg-

mentation task in terms of the structures of input and output.

4.2 Advanced Next Conversation Prediction

The predictive difficulty of the negative samples (label 0) in NCP depends on the distance between the left and right contexts, l and r, in the transcript. A greater distance between the contexts makes the NCP task easier but more different from, and hence potentially less useful for, the segmentation task. Similarly, the difficulty of the positive samples (label 1) depends on how semantically similar the contexts are to each other. Positive samples with highly semantically similar contexts will be easy to identify. To control the difficulty of the NCP samples, we introduce the following conditions on the positive and negative samples of the pretraining data respectively.

$$Sim(l, r) \geq s \text{ for label } 1 \quad (1)$$

$$\text{Dist}(l, r) = d \text{ for label } 0 \quad (2)$$

where $Sim()$ is a semantic similarity function [1] quantifying similarity between l and r, and s is the similarity coefficient. $\text{Dist}()$ is the distance between l and r in terms of number of sentences between them and d is the distance coefficient. Figure 2 illustrates the Advanced NCP.

In vanilla NCP, by default, the distance between non consecutive contexts is greater than 1 and there is no semantic similarity filter ($s = 0$). Increasing s will make the pretraining task easier as the positive samples (label 1) have the additional constraint of being semantically similar. However, increasing s too much can filter out too much pretraining data. Similarly, decreasing d will make the task harder but more relevant to the segmentation task.

4.3 Curricular Next Conversation Prediction

Curriculum learning (Bengio et al., 2009) proposes that models observing training samples in an increasing order of difficulty have an advantage over models observing samples in an otherwise random order. Motivated by this, we introduce a curriculum to Advanced NCP pretraining. The pretraining samples from Advanced NCP are divided into two distinct sets - "similar" to downstream task (or "harder" since, in general, segmentation is a harder task than NCP) and "dissimilar" to the segmentation task (or "easier"). In order to estimate the similarity or dissimilarity of the NCP pretraining samples to the segmentation task, we use a classification model trained for the segmentation task and use it to predict labels for the pretraining instances. We refer to this as the "Auxiliary model" and classify a pretraining sample as "similar" if the Auxiliary model correctly predicts its label and vice versa. In the spirit of curriculum learning, we divide the pretraining into two steps. First training on the "dissimilar" or "easy" (from the perspective of segmentation) samples followed by the "similar" or "hard" samples. This order makes sure that the model has smoother transition between the two tasks that are semantically close but different. Figure 6 illustrates the Curricular NCP process. Table 4 shows examples of Dissimilar NCP and Similar NCP from the SliceCast-Podcast dataset. All the examples are labeled 0 in their respective tasks.

While the Auxiliary Model can be any classification model trained on the segmentation task dataset, we use a model that is additionally pretrained on Advanced NCP data. The Auxiliary model is tested on Advanced NCP samples. While these samples were used during the pretraining of the Auxiliary model, after finetuning on the segmentation task model might not predict the same labels it observed during the pretraining. In our experiments with the SliceCast-Podcast dataset (described in Section 5.1) we indeed observe that 64.4% samples are miss-classified (hence, "dissimilar") and 35.6% are correctly classified (hence, "similar").

5 Experimental details

In this section, we describe the dataset, the base model upon which our pretraining is tested, the implementational details, metrics and baselines.

5.1 Dataset

We use the SliceCast-Podcast (Midei and Mandic, 2019) dataset for our experiments. This dataset has 46 podcast transcripts and a total of 643 segments. On average, each transcript has 12.4 segments, though there could be high variation in number of segments as the standard deviation is 4.1. We consider 416 segments for training and 181 segments for testing purposes. While creating training data for the pretraining and the segmentation task, positive and negative samples are sampled equally. For this, in the segmentation task we randomly down sample samples labeled 1. Figure 1 shows examples of segment breaks from this dataset.

[1] We use Sentence BERT (Reimers and Gurevych, 2019) to compute representations of l and r, followed by cosine similarity for $Sim()$

5.2 Base Binary Classification Model

We use BERT as the base classification model in the Auxiliary model. Across all classification tasks - Advanced NCP, Curricular NCP and the Segmentation task, the input is provided by concatenating l and r contexts with the token "[SEP]". Hence the input is " l [SEP] r ". T, the maximum input size of l and r individually is 150[2]. The output is taken from the first position ("[CLS]") and a binary cross entropy layer is attached to enable binary classification.

Lukasik et al. (2020) propose Cross Segment BERT, for the transcript segmentation task. We use this as the base segmentation model for transcript segmentation after finetuning it on the data described in Section 5.1.

5.3 Coefficient details for Advanced NCP

The two coefficients explained in Section 4.2 control difficulty of the Advanced NCP task and hence its relevance to the pretraining task. We experimented with various values of s and d in the Advanced NCP task. Following which, we measure performance of the these pretrained models (with different values of s and d) on a balanced Advanced NCP test data. The accuracy results are reported in Fig. 3a. A darker shade of green indicates better performance. As we can see, in general, Advanced NCP performance (accuracy) increases as the coefficients increase, making the positive and negative samples easier to identify. However, large values of s results in filtering out too many positive samples and hence the size of the training dataset leading to a decreasing in performance.

The aforementioned Advanced NCP pretrained models are then fine-tuned on the segmentation task. All the models are finetuned on the dataset described in Section 5.1. For model comparison we use the F1 of the the segment break class (0) on a held out set of containing 61 segments. Results of the finetuned models are illustrated in Figure 3b. Comparing Figures 3a and 3b we can observe that while a low performance on the Advanced NCP task also corresponds to a low performance on the segmentation task, the converse is not true. At high coefficient values, especially the distance coefficient d, the pretraining task is too distinct from the segmentation task leading to low efficacy of the pretraining strategy.

[2]Original Cross Segment BERT (Lukasik et al., 2020) used 125 in most of their experiments. We follow a similar setting

Models	F1 (↑)	Pk (↓)	WDiff (↓)
S-BERT	4.1	50.5	65.1
CSB	17.5	42.5	37.3
Adv. NCP + CSB	22.1**	37.2**	**36.2****
Curr. NCP + CSB	**22.6***	**35.6****	37.5**

Table 1: Evaluation results for S-BERT (Solbiati et al., 2021), CSB (Lukasik et al., 2020) and CSB pretrained with the proposed strategies. WDiff refers to WindowDiff. Pretrained models significantly outperform CSB in all metrics. Introduction of curriculum to Advanced NCP also shows improvement. * and ** denote the difference is significant with $p < 0.03$ and $p < 0.06$ via t-test.

Models	F1-0 (↑)	Pk (↓)	WDiff (↓)
Hier.	19.9	39.2	37.1
Adv. NCP + Hier.	**20.6**	38.5	**36.2**
Curr. NCP + Hier.	20.3	**37.4**	36.6

Table 2: Performance of the Hierarchical (Hier.) model (Lukasik et al., 2020) before and after pretraining with Advanced NCP (Adv. NCP + Hier.) and Curricular NCP (Curr. NCP + Hier.). Application of pretraining on Hierarchical shows improvement.

Using these two figures, we find the ideal coefficients such that the Advanced NCP strategy is sufficiently easy but relevant to the downstream task concurrently. We choose $d = 200$ and $s = 0.7$.

5.4 Metrics

In line with previous works (Midei and Mandic, 2019; Lukasik et al., 2020; Solbiati et al., 2021), we use F1 score and Pk score (Beeferman et al., 1999) for evaluating text segmentation models. The scores are calculated by using the model to predict existence of segment break after each sentence end in the test set and then comparing ground truth segment break predictions and predicted segment breaks. F1 score of the label 0 is considered. This score is a strict measure as it rewards the model only if the predicted segment breaks and ground truth segment breaks exactly align. Pk score is less harsh. Pk score is calculated by using a sliding window such that predicted segment breaks near ground truth are penalised less than predictions that are far away[3]. One criticism of the Pk score is that it favours models that make fewer segment break predictions. To address this Pevzner and Hearst (2002) proposed WindowDiff to account for the number of segment break predictions as well. For WindowDiff and Pk, we consider size of sliding window to be half of the average segment length in number of sentences, as is the standard practice.

[3]we encourage the reader to look at assemblyai.com

s ↓ d →	15	66	100	200	500
0.4	0.58	0.63	0.61	0.69	0.79
0.6	0.61	0.69	0.59	0.77	0.78
0.7	0.59	0.75	0.69	0.77	0.8
0.8	0.62	0.74	0.76	0.82	**0.89**
0.9	0.59	0.71	0.68	0.7	0.68

(a) Advanced NCP task performance in Accuracy. A higher number implies low task difficulty.

s ↓ d →	15	66	100	200	500
0.4	0	0.04	0.1	0.12	0.12
0.6	0.001	0.11	0.11	0.14	0.1
0.7	0.001	0	0.17	**0.24**	0.09
0.8	0.02	0.12	0.18	0.18	0.11
0.9	0.03	0.04	0.06	0	0.09

(b) Segmentation task performance in F1 score of the 0 label across different Advanced NCP pretraining.

Figure 3: Performance of Advanced NCP pretraining and Segmentation task with varying coefficients d and s. As we can observe higher pretraining task performance does not necessarily imply higher downstream task performance.

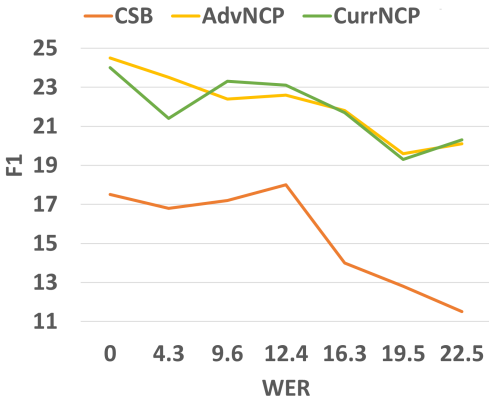

Figure 4: Performance with respect to change in transcription errors (WER). x-axis is represents different WER rates and y-axis represents the F1 scores. Proposed pretraining strategies makes the CSB model more robust to transcription errors.

For both Pk and WindowDiff, lower scores indicate better performance.

5.5 Baselines

S-BERT We compare the supervised techniques (pretrained and non pretrained) against this Sentence BERT based unsupervised transcript segmentation baseline (Solbiati et al., 2021) to show the motivation for this task to be addressed in a supervised setting.

Cross Segment BERT (CSB) This is a BERT model for the segmentation task without any proposed pretraining. The model is based on Lukasik et al. (2020), originally proposed for document segmentation where left and right contexts are concatenated with a separator and provided to the BERT model for binary classification. CSB formed a state-of-the-art in text segmentation. Hence, we apply our pretraining strategies on this model.

6 Results and Discussions

6.1 Pretraining on Cross Segment BERT

Table 1 presents the results of application of the proposed pretraining strategies - Advanced NCP and Curricular NCP on CSB (Advanced NCP + CSB and Curricular NCP + CSB respectively). We also compare with S-BERT, the unsupervised baseline.

Challenges of an unsupervised setting The unsupervised baseline, S-BERT, vastly underperforms all other models (supervised) on all metrics (row 1 and other rows). This is because the definition of a segment could be data and domain specific. In such a case, deriving its interpretation from a supervised data becomes imperative. Hence, despite the difficulty of annotation, supervised approaches are favoured.

Improvement due to proposed pretraining By comparing pretrained models with CSB (row 2 and rows 3,4), we see that pretrained models outperform across all metrics indicating their effectiveness. We also outperform the transcript segmentation baseline proposed by Midei and Mandic (2019). However, we do not apply our pretraining strategy to it since it adopts a sequence labeling paradigm, and adapting proposed pretraining strategies for such models is left for future work.

By comparing Advanced NCP and Curricular NCP (row 3 and row 4), we see that proposing a curriculum to the pretraining leads to better F1 and Pk scores. We give two major reasons for this improvement - First, our ordering of pretraining samples in Curricular NCP is relevant to the segmentation task. Prior research in curriculum learning show such sample orderings are more effective than arbitrary sample orderings such as sentence length for sentiment analysis (Rao et al., 2020). Second, the transcript segmentation data is small in size and previous works note the efficacy of curriculum learning in resource poor settings (Cirik et al., 2016; Nagatsuka et al., 2021).

6.2 Pretraining on Sequence Labeling

To further observe efficacy of the proposed pretraining approaches, we apply them on a sequence labeling approach. We use a model based on Hierarchical BERT model from Lukasik et al. (2020) which is compatible with our pretraining task. In

this "Hierarchical" baseline- first, the left and the right context pairs are obtained by taking $T = 150$ tokens of left and right context around each sentence end. Next, CSB model is used to obtain representations for context pairs. Hence, each transcript is converted to sequence of context pair representations. Finally, this sequence of context pair representations is then input to an LSTM (50 units) in a one-to-one sequence labelling setting to output an equally long sequence of 1s and 0s. Similar to segmentation in the binary classification setting (explained in Section 3), 0 indicates a segment break and 1 indicates absence of a segment break. In this hierarchical baseline, we swap the CSB model with Advanced NCP + CSB model to obtain Advanced NCP + Hierarchical model and using a similar process we obtain Curricular NCP + Hierarchical.

The performances of all models are reported in Table 2. By comparing the pretrained models (Advanced NCP + Hierarchical and Curricular NCP + Hierarchical) to the model without pretraining (Hierarchical), we observe an improvement. The performance increment between vanilla and pretrained models, has diminished slightly in this sequence labelling setting as compared to CSB as based model setting. This is possibly because the Hierarchical model, involves more parameters (LSTM units) that have not been updated during our pretraining steps as opposed to the CSB model, where all parameters were involved in the pretraining. Regardless, pretraining leads to an improvement across all metrics. This is consistent with Table 1, showing that proposed pretraining methods have merits across the downstream model architecture (CSB or Hierarchical).

6.3 Utility in an unsupervised setting

To further understand the relationship between the pretraining and the segmentation task, we do cross domain testing. Here, we use an NCP pretrained model (prior to finetuning) to make predictions on the segmentation test data. Since NCP does not use any segmentation information, this method is unsupervised in segmentation prediction. We also make predictions on Curricular NCP pretraining test data using the segmentation model ("Pretraining Test Data").

Results of this experiment are tabulated in Table 3. For the pretraining test data, we use accuracy for performance comparison. WindowDiff is used for the segmentation test data. Comparing the performances of S-BERT and NCP on the segmentation task, we observe that NCP outperforms S-BERT. This shows that the proposed pretraining approximates the segmentation task and gives the necessary domain knowledge to perform well even in an unsupervised setting. Next we compare the performances of the models trained for segmentation task (Curricular NCP + CSB and CSB) with the performance of NCP. We can see that Curr NCP +CSB and CSB are performing better than NCP on segmentation task but not on pretraining task. This shows that is a significant difference between the two tasks.

Models	**Pretraining Test Data** (↑)	**Segmentation Test Data** (↓)
S-BERT	55.8	65.1
NCP	69.3	61.5
CSB	61.8	37.3
Curr. NCP + CSB	66.4	37.5

Table 3: Results of the cross domain testing experiment. We report accuracy for the pretraining task and WindowDiff for the segmentation task. NCP does not use any segment information and outperforms S-BERT in segmentation, thereby forming a strong unsupervised approach.

6.4 Robustness

Since, automatically generated transcripts tend to be noisy, in this section we measure the robustness of proposed pretraining strategies to noise in training data. In this experiment, we synthesize noise in the training samples using Easy Data Augmentation (EDA) (Wei and Zou, 2019). EDA introduces noise to transcript samples by four operations - synonym replacement, random insertion, random swap, and random deletion. EDA also provides a temperature variable to control how intensely these operations are applied. By increasing the temperature in some of these operations, we obtain six SliceCast-Podcast variants with increasing WER rates (4.33%, 9.60%, 12.42%, 16.33%, 19.51% and 22.53%) with respect to the original dataset. Only random insertion, swap, and deletion are used for introducing noise. Note that the test data is not changed across these variants. Figure 4 shows the performance (F1) of CSB, and CSB model pretrained with Advanced and Curricular NCP.

We observe that the results align with the results reported in Table 1 i.e. First, Pretrained models always outperform CSB. Second, Curricular NCP pretrained model generally outperforms Advanced

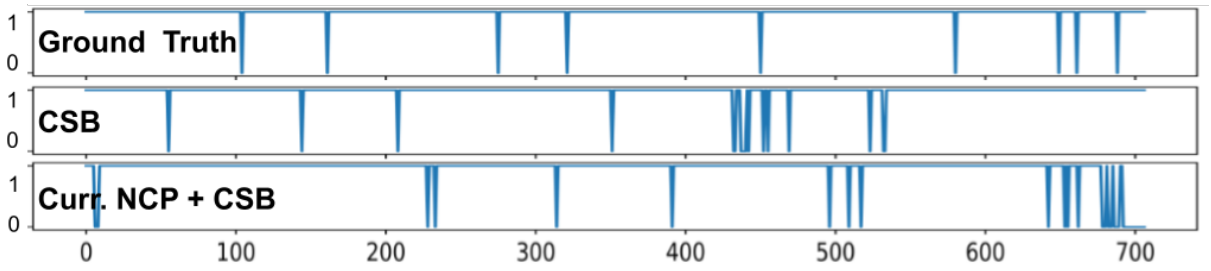

Figure 5: Segmentation break predictions across all sentences of a test transcript, illustrated in the ground truth annotation, annotation by CSB model and annotation by Curricular NCP pretrained CSB model. The pretrained model is able to catch the highly segmented area of the transcript.

NCP pretrained model. Furthermore, we observe a decreasing performance trend in all the models as WER increases. This is expected behaviour because as the data becomes more noisy, we loose valuable clues that reflect start and end of a segment. However, we observe that the pretrained models have a lower decrement in performance as compared to CSB as the WER increases. The overall performance decrement among CSB, Advanced NCP + CSB and Curricular NCP + CSB is -5.7%, -1.7%, -2.4% respectively. This shows that pretraining introduces robustness in the model.

6.5 Qualitative Analysis

To further analyze the advantages of pretraining, we visualize the segment break predictions across all sentences of a transcript from the test set. Figure 5 shows segment break annotations in the ground truth, and predictions by the CSB model and the Curricular NSP pretrained CSB model. x-axis represents the number of sentences and y-axis represents label predictions. As discussed in Section 4.2, we use the model after each sentence to predict segment breaks. Looking at the ground truth annotations, we can see that segment lengths can vary greatly within a transcript. Some segments are more dense than others. We can observe that Curricular NCP helps the model to correctly identify a region of dense segment breaks. Identifying such dense regions might require large training data to correctly understand the dynamics of segment sizes. In such cases, pretraining of NCP can make up for less labelled data.

6.6 Error Analysis

We further investigate the kind of errors the models (with and without pretraining) are making. In general, we note both CSB and pretrained CSB tend to over-predict segment breaks. Their precision and recall for label 0 are as follows - 14.6 and 22.1 for CSB and 20.6 and 25.1 for Curricular NCP + CSB. This is consistent with Figure 5 where we observe that pretrained model is better at identifying highly segmented areas.

Next, we manually analyzed the kinds of errors the models are making. We find that both models over-rely on certain cues to over-predict segment breaks. For example, the models, with and without pretraining, were more likely to predict a segment break for samples in which the left context ended in a question but the ground truth data had no such bias. Similarly, among CSB's segment break predictions, 8.29% had "yeah" in the beginning of the right context, whereas this number is only 6.63% in the ground truth segment breaks. Pretraining reduces this over-reliance (the corresponding number for Curricular NCP + CSB is 6.84%). Tables 5 and 6 provide more information. We leave further investigations into these errors for future work.

7 Conclusion

In this paper, we propose novel pretraining strategies for transcript segmentation. Our pretraining strategies address major challenges associated with transcript data. The pretraining strategies are based on the idea of next conversation prediction. This strategy by itself also forms a strong unsupervised baseline for segmentation. Additional improvements make NCP more relevant and useful to the segmentation task. We further introduced a curriculum in the pretraining strategies based on similarity of pretraining samples to the segmentation samples. Our results showed that our proposed pretraining strategies are robust to noise in training data and they are effective for improving performance of multiple model architectures for segmentation.

8 Limitations

NCP requires the dataset to be marked with sentence breaks. Segmentation datasets might not have this annotation. While an off-the-shelf sentence break identifier model can do this sub-task, this could introduce some noise to the training dataset.

While we have shown that NCP applies to multiple segmentation task architectures (Hierarchical and CSB in Tables 1 & 2), it might not be applicable across all segmentation architectures. Since NCP relies on its similarity to the segmentation task, pretraining on differently defined segmentation tasks might not yield benefits without alterations.

A different transcript segmentation dataset might be significantly different from NCP such that the pretraining's benefits taper off. However, it is hard to comment on this with the currently available datasets for this task.

We hope that future work explores these concerns and that our work can be a stepping stone in this exciting direction.

9 Ethical Considerations

We train our model on a publicly available podcast dataset that might contain (potentially harmful) social biases. Furthermore, since this an informal use of language, the text is rife with colloquialisms, some of which could be triggering or sexually explicit. Since, we have not employed any bias removal methods, model might predict segment breaks based on spurious correlations such as usage of specific pronouns or mention of specific genders. All the trained models are only tested on English language dataset and might not necessarily carry well to other languages.

References

Jaime Arguello and Carolyn Rosé. 2006. Topic segmentation of dialogue. Analyzing Conversations in Text and Speech (ACTS), pages 42–49.

Pinkesh Badjatiya, Litton J Kurisinkel, Manish Gupta, and Vasudeva Varma. 2018. Attention-based neural text segmentation. In European Conference on Information Retrieval, pages 180–193. Springer.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. Machine learning, 34(1):177–210.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In Proceedings of the 26th annual international conference on machine learning, pages 41–48.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. arXiv preprint arXiv:1803.11175.

Freddy YY Choi. 2000. Advances in domain independent linear text segmentation. In 1st Meeting of the North American Chapter of the Association for Computational Linguistics.

Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. arXiv preprint arXiv:1611.06204.

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.

Jacob Eisenstein. 2009. Hierarchical text segmentation from multi-scale lexical cohesion. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 353–361.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2016. Unsupervised text segmentation using semantic relatedness graphs. In Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics, pages 125–130, Berlin, Germany. Association for Computational Linguistics.

Alexander Gruenstein, John Niekrasz, and Matthew Purver. 2008. Meeting structure annotation. In Recent Trends in Discourse and Dialogue, pages 247–274. Springer.

Marti A Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. Computational linguistics, 23(1):33–64.

Pei-Yun Hsueh, Johanna D Moore, and Steve Renals. 2006. Automatic segmentation of multiparty dialogue. In 11th Conference of the European Chapter of the Association for Computational Linguistics, pages 273–280.

Elise Jing, Kristiana Schneck, Dennis Egan, and Scott A Waterman. 2021. Identifying introductions in podcast episodes from automatically generated transcripts. arXiv preprint arXiv:2110.07096.

Omri Koshorek and Adir Cohen. 2017. Learning text segmentation using deep lstm.

Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. 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 469–473.

Jing Li, Aixin Sun, and Shafiq R Joty. 2018. Segbot: A generic neural text segmentation model with pointer network. In IJCAI.

Cao Liu, Shizhu He, Kang Liu, Jun Zhao, et al. 2018. Curriculum learning for natural answer generation. In IJCAI, pages 4223–4229.

Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. 2020. Text segmentation by cross segment attention. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4707–4716, Online. Association for Computational Linguistics.

Brian Midei and Marko Mandic. 2019. Neural text segmentation on podcast transcripts. github.com/bmmidei/SliceCast.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2021. Pre-training a bert with curriculum learning by increasing block-size of input text. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), pages 989–996.

Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics, 28(1):19–36.

Vijjini Anvesh Rao, Kaveri Anuranjana, and Radhika Mamidi. 2020. A sentiwordnet strategy for curriculum learning in sentiment analysis. In International Conference on Applications of Natural Language to Information Systems, pages 170–178. Springer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 453–463.

Alessandro Solbiati, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. 2021. Unsupervised topic segmentation of meetings with bert embeddings. arXiv preprint arXiv:2106.12978.

Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In Proceedings of the 39th annual meeting of the Association for Computational Linguistics, pages 499–506.

Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020. Curriculum pre-training for end-to-end speech translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3728–3738.

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.

Jinxiong Xia, Cao Liu, Jiansong Chen, Yuchen Li, Fan Yang, Xunliang Cai, Guanglu Wan, and Houfeng Wang. 2022. Dialogue topic segmentation via parallel extraction network with neighbor smoothing. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2126–2131.

Linzi Xing and Giuseppe Carenini. 2021. Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring. In Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 167–177.

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

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2019. Outline generation: Understanding the inherent content structure of documents. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 745–754.

A Appendix

A.1 Additional information on the Text Segmentation task

Transcript segments represent a mix of change in topics, sub-topics and/or nature of discourse. For example, new segments may start when the participants change their discussion from Health domain to Toastmasters or within health from mammograms to genetics. Other ways segments may change based on whether the discussion has changed from a short dialogue style conversation

Pretraining Example “Disimilar” to Segmentation Task	[...] get serious. so i think it’s appropriate that this is the week that we’re going to talk about don’t just sit there. right. in this episode of the podcast. yes yes. so great.",[SEP] ’tell me a little bit about how this book came to be. oh, this book was written right after move your dna. like six weeks after. and i wrote it because mark sisson who is a big paleo icon and has... primal blueprint is his big book. he wanted me to write, [...]
Pretraining Example “Similar” to Segmentation Task	[...] spinner is. only to realize that it’s a thing that everyone else knows except for me. right well you weren’t on social media all summer so that’s how that got by you.",[SEP] ’maybe but even my kids didnt́ know what they were and then they went to a birthday party where everyone else had them and they were like, " we have to have fidget spinners. " [...]
Downstream Task Example	[...] so thinking about writing those letters. like there’s the difference in calling, maybe, there’s something in it for the writer as well. yeah. you encounter yourself in a different way." [SEP] "at least that’s my experience as a writer. when i am on the page with words in my hand, moving across a piece of paper, i’m writing to whoever i’m writing to. [...]

Table 4: Examples of “similar” and “dissimilar” samples to the downstream task. The ordering from top to bottom is also the order we follow for training Curricular NCP.

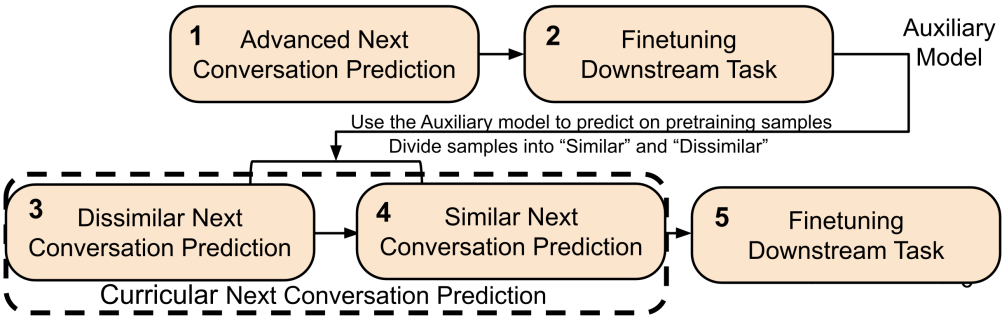

Figure 6: Proposed Curricular NCP Pretraining illustrated. First the auxiliary model is obtained to rank the pretraining samples into “similar” and “dissimilar”. Following which the curriculum can be followed.

Models	SB	Not SB
CSB	5.07	1.38
Curr. NSP + CSB	4.56	1.01
Ground	3.87	3.31

Table 5: Percentage of samples which had a "?" within the last 10 characters of the left context. Here, “Segment Break” (SB) and “Not Segment Break” (Not SB) implies ground truth in “Ground” and predictions in case of the models. For example, 5.07% of samples predicted with a segment break for CSB had left context ending in “?”. Both CSB and Curricular NSP + CSB tend to over predict segment breaks when the left context ends with a "?" compared to ground truth, which has no such bias.

Models	SB	Not SB
CSB	8.29	0.69
Curr. NSP + CSB	6.84	1.01
Ground	6.63	3.87

Table 6: Percentage of samples which had "yeah" within the first 5 words of the right context. Here, “Segment Break” (SB) and “Not Segment Break” (Not SB) implies ground truth in “Ground” and predictions in case of the models. For example, 6.84% of samples predicted with a segment break for the pretrained model had “yeah” within five tokens after the predicted segment break. While both CSB and Curricular NSP + CSB make incorrect predictions, the distribution is more closer to ground truth after pretraining.

to a long answer QA session. The individual segments are often too verbose and diverse (average length 206.5 words and standard deviation 500.13) to be presented unedited. Hence, we gave an idea of what these segments look like in Figure 1, with individual sentences of a segment truncated.

A.2 Toolkits

We use NLTK toolkit Link: https://www.nltk.org/ for computing WindowDiff https://www.nltk.org/_modules/nltk/metrics/windowdiff.html. NLTK version is 3.6.2.

A.3 Training and Inference Details

Number of parameters:BERT-base has 110 million parameters.

GPU Details: We use a NVIDIA GeForce RTX 2080 Ti machine to train and infer all our models. All experimental results except for Tables 6 and Tables 5 are reported over a mean of 3 runs.

A.4 Dataset License Details

The dataset we have used SliceCast-Podcasts, Link - https://github.com/bmmidei/SliceCast#Small-scale-podcast-dataset was licensed under the MIT License. Our research is consistent with the intended use.