Topic Segmentation of Recorded Meetings

by

Ilir Lazoja

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2024

© Ilir Lazoja 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Video chapters allow videos to be more easily digestible and can be an important pre-processing step for other video-processing tasks. In many cases, the creator can easily chapter their own videos, especially for well-edited structured videos. However, some types of videos, such as recorded meetings, are more loosely structured with less obvious breaks, which makes them more cumbersome to chapter and thus would highly benefit from being automated. One approach to chaptering these types of videos is through performing topic segmentation on the transcript of the video, especially if the video is rich in dialogue. Topic segmentation is the task of dividing text based on when the topic of the text changes, most commonly performed on large bodies of written text. This thesis will detail how well state-of-the-art approaches for topic segmentation performs on recorded meetings, as well as present and evaluate strategies to improve performance for recorded meetings and express shortcomings of the common metrics used for topic segmentation.

Acknowledgements

I would like to thank my supervisors and my family for the support, as well as colleagues at the VIP Lab for motivation and advice.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

List of Figures viii

List of Tables x

1 Introduction 1

2 Related Work 3

2.1 Existing Datasets . . . 3

2.2 Evaluation . . . 4

2.2.1 P_k and WindowDiff . . . 5

2.2.2 *MIoU* and *EIoU* . . . 6

2.3 Techniques . . . 8

2.3.1 Literature . . . 8

2.3.2 Commercial . . . 10

3 Experimental Setup **13**
3.1 Datasets . . . 13
3.1.1 Pre-Processing . . . 13
3.1.2 Custom YouTube Dataset . . . 14
3.1.3 Summary and Statistics of Datasets . . . 16
3.2 Model Architectures . . . 17
3.2.1 SBERT LSTM . . . 17
3.2.2 Cross-Segment BERT . . . 19
3.2.3 Cross-Segment BERT LSTM . . . 20
3.2.4 SBERT Text-Tiling . . . 20
3.3 Loss Function . . . 21
3.4 Label Smoothing . . . 23
3.5 Augmentation . . . 24
3.6 Ensemble . . . 24
3.7 ChatGPT . . . 24
3.8 Summary . . . 26

4 Observations **29**
4.1 Effect of Different Models . . . 29
4.2 Effect of Transfer Learning . . . 31
4.3 Effect of Label Smoothing and Augmentation . . . 35
4.4 Effect of Ensemble . . . 35
4.5 Run Time . . . 36
4.6 Critique of Metrics and Suggestion for an Improved Metric . . . 39

5 Conclusions and Future Work **42**

References **44**

APPENDICES **50**

A Chapters in a Sample Transcript **51**

List of Figures

2.1 A visual illustration of the sliding window evaluation for the P_k and *WD* metric. . . . 6

2.2 A visual illustration of superfluous prediction, effected prediction, and missed prediction. . . . 7

2.3 General formulation of text segmentation as a supervised learning task. . . . 8

2.4 Alternate formulation of text segmentation as a supervised learning task. . . . 9

2.5 Example of a cosine similarity comparison for an unsupervised approach to topic segmentation. . . . 10

2.6 Examples of videos automatically chaptered by YouTube. . . . 12

3.1 Raw sample of AMI meeting IS1002d. . . . 14

3.2 Pre-processed sample of AMI meeting IS1002d for topic segmentation. . . . 15

3.3 Example of a manually chaptered YouTube video. . . . 15

3.4 The SBERT LSTM model for chaptering. . . . 18

3.5 The Cross-Segment BERT model for chaptering. . . . 19

3.6 The Cross-Segment BERT LSTM model for chaptering. . . . 21

3.7 SBERT Text-Tiling approach. . . . 22

3.8 Pre-processed sample of AMI meeting IS1003a for topic segmentation. . . . 23

3.9 A sample augmentation. . . . 25

3.10 Ensemble bagging methodology. . . . 26

3.11 Sample ensemble bagging example. . . . 27

3.12 A sample query to ChatGPT. 28

4.1 Predictions of each chaptering method on sample meeting TS3007a. 31

4.2 SBERT LSTM training and validation curves for each metric across epochs. 32

4.3 Predictions on sample meeting TS3007a after transfer learning on various datasets. 34

4.4 SBERT LSTM with augmentation, focal loss, and label smoothing training and validation curves for $EIoU$ across epochs. 35

4.5 Predictions on sample meeting TS3007a after employing label smoothing, augmentation and focal loss. 37

4.6 Chaptering predictions with ensemble classification on sample meeting TS3007a. 38

4.7 A visual illustration of the sliding window evaluation for the P_k and WD metric when window size k is 1. The same amount of error is produced as in Figure 2.1 regardless of window size. 39

4.8 A visual illustration of the revised window evaluation. 40

List of Tables

3.1 Summary and statistics of datasets. 17

4.1 Performance of different chaptering methods on AMI dataset. 29

4.2 Chaptering performance of train, val, and test sets after training on Wiki-727k and YT-Programming. 33

4.3 Chaptering performance of transfer learning on AMI dataset. 33

4.4 Performance of augmentation, label smoothing, and focal loss on AMI dataset. 36

4.5 Inter-domain performance after training on Wiki-727k and YT-Programming with label smoothing. 36

4.6 Performance of ensemble classification on AMI dataset. 36

4.7 Run time of different chaptering methods on AMI training dataset. 38

4.8 Performance of different chaptering methods on AMI dataset with revised metrics. 41

Chapter 1

Introduction

Videos are increasingly becoming one of the main ways we consume media and receive information in today's age. Unfortunately, compared to text, it is more difficult to sift through the video for specific information that is desired. Video platforms have introduced chapters navigable through the scrubber to help users get the information they need. Typically, the creator of the video will chapter videos according to how they structured their video, however not every video is neatly structured in a way that would make chaptering easy to perform. Such is the case for recorded meetings, a common type of video in the business space. Chaptering a meeting manually would require an individual to review the meeting themselves, which is not usually worth the time investment. Automated chaptering can provide huge value to recorded meetings as it would allow individuals to quickly be able to retrieve the information they need, whether it be to review a meeting they missed or to refresh themselves on the topics discussed, providing a significant boost in efficiency to the individual and the overall business.

Meetings are rich in dialogue, but often dry in visual information, and thus meeting chaptering can be considered to fall under the natural language processing (NLP) problem area of topic segmentation, the task of dividing text when a change of topic occurs. In contrast, other types of videos, such as movies or documentaries, that have richer visual information would fit the scene segmentation problem better, which can take into account the multiple modalities of the video [28] [34], at the cost of computation resources. Most topic segmentation research focuses on segmenting written text, with few that focus on segmenting transcripts from spoken dialogue [11]. Dialogues tend to be more unstructured due to the natural and spontaneous way a conversation between individuals can flow, as well as be more informal in how words are spoken [11]. Meetings often consist of frequent

dialogue between individuals, although there may be prolonged moments of monologue if an individual is giving a presentation.

Current state-of-the-art methods for topic segmentation encode each sentence using transformers and use supervised learning to learn the relationship between sentences and a topic start [21] [23] [37]. This thesis will investigate how the state-of-the-art methods in topic segmentation perform on meeting transcripts, identify the challenges with meeting chaptering, and experiment with strategies to boost performance. The key contributions are as follows:

1. Recent approaches of topic segmentation proposed for segmenting written text (Section 3.2) are compared on public meeting datasets (Section 4.1).

2. The effectiveness of domain transfer is evaluated by pre-training on different datasets of various domains (Section 3.1).

3. A new dataset is formed, acquired from YouTube tutorial videos, for the purpose of pre-training (Section 3.1.2). Its effectiveness is evaluated compared to pre-training on established written text datasets (Section 4.2).

4. An ensemble approach is introduced for the task, which mimics the subjective nature of ground-truth labels (Section 3.6). Augmentation (Section 3.5) and label smoothing (Section 3.4) were also employed, however, both techniques were ineffective (Section 4.3).

5. Current metrics used for topic segmentation are critiqued and revised metrics are suggested (Section 4.6).

Chapter 2

Related Work

2.1 Existing Datasets

There are three datasets publicly available that contain labelled segments of meeting videos as well as transcription: the AMI Meeting Corpus [26], the ICSI Meeting Corpus [19], and QMSum [40].

The AMI Meeting Corpus contains recordings of two different categories of videos: real meetings and mock corporate meetings, the latter of which contains most of the chapter annotations [26]. In the mock corporate meetings, participants assume a role at an electronics company and are tasked with discussing a new design for a remote control [26]. The structure of an individual meeting falls under one of four categories mimicking the design process: project kick-off, functional design, conceptual design, and detailed design [26]. The annotators were instructed to chapter the meetings in a way they found most 'natural' [38].

The ICSI meeting videos are recorded from real research group meetings held at the International Computer Science Institute (ICSI) in Berkeley, California [19]. The types of meetings recorded include discussing NLP, the ICSI meeting corpus itself, robustness in speech recognition, network services as well as some miscellaneous topics [19]. The original dataset had few topic segmentation labels, additional videos were annotated by Gruenstein et al. [14]. The annotators were instructed to chapter meetings based on the utility of being able to quickly browse portions of the video that they would be interested in reviewing [14].

QMSum combines the AMI and ICSI datasets as well as features a small new set of chaptered committee meetings from the Welsh Parliament and Parliament of Canada [40].

Annotators were instructed to chapter only the most important main topics, with a range of around 3 to 8 chapters per meeting [40].

AMI and ICSI datasets are chaptered with a hierarchy (i.e. have nested sub-chapters within chapters), which gives the opportunity to choose whether to use only the top-level chapters or flatten the hierarchies, treating each sub-section of the hierarchies as full separate chapters. In general, these meeting datasets are sufficient for evaluation but do not contain many videos for training.

Besides these meeting datasets, there do not seem to be any larger dialogue datasets created that have deliberate topic segmentation annotations. Some researchers utilize existing dialogue datasets that do not have topic segmentation and create synthetic topic segmentation labels by leveraging other data points or rearranging the dialogue. For example, Ghosh et al. [11] created synthetic topic segmentation labels from the Amazon Topical Chat [13] and the LDC BOLT SMS/Chat [33] datasets, which feature text conversations between individuals, by considering the start and end of a conversation as a topic segment. Xing et al. [37] use Doc2Dial, a dataset that maps auto-generated and human dialogue to documents in government service websites, and considers the dialogues mapped to corresponding sections in a document as a chapter. Yi et al. [39] leverages the MultiWOZ corpus [4] and the Stanford Dialog Dataset [10] where each dialogue discusses a single topic.

There are several datasets available for topic segmentation of written text. The largest of which is Wiki-727K, consisting of English Wikipedia articles [21]. Wikipedia articles are hierarchically segmented with topics and sub-topics which lends itself well to training and evaluating topic segmentation. Another related noteworthy dataset is VT-SSUM, which uses the mapping of the transcribed lecture to slides from VideoLectures.NET to determine segment labels (i.e. a change in the slide is considered a new topic) [24].

2.2 Evaluation

Topic segmentation is performed by labelling whether a sentence is the start of a change in topic in a text [21]. Naturally, an accuracy measurement would not be a helpful metric, since there are fewer sentences that are topic boundaries than not. Precision, Recall, and F1 score are better metrics for this type of problem, although windowing needs to be implemented so that near-misses are not penalized since predicted topic boundaries that are close enough should still be rewarded. The issue with introducing a window is the window length is arbitrary. Thus, alternate metrics are used for the problem of topic segmentation.

2.2.1 P_k and WindowDiff

P_k [3] and WindowDiff (WD) [27] are common error metrics that are used for topic segmentation [25] [32] [36]. Both are evaluated assuming each sentence is labelled 0 or 1, where 1 represents a topic change boundary, using a sliding window as shown in Figure 2.1. P_k is evaluated using the following equation:

$$P_k = \frac{1}{N-k}\sum_{i=1}^{N-k}((b(y_i, y_{i+k}) > 0) \neq (b(p_i, p_{i+k}) > 0)) \tag{2.1}$$

Whereas, WD is evaluated using the following:

$$WD = \frac{1}{N-k}\sum_{i=1}^{N-k}(|b(y_i, y_{i+k}) - b(p_i, p_{i+k})| > 0) \tag{2.2}$$

where,

$$k = \frac{N}{2b(y_{0,N})} \tag{2.3}$$

and N is the number of sentences, $b(y_i, y_j)$ is the number of ground-truth boundaries between sentences i and j, and $b(p_i, p_j)$ is the number of predicted boundaries between sentences i and j. Both metrics calculate the average error from each sentence index, thus a low score is desired for both .

A notable limitation with P_k which WD addresses is it does not take into account the number of sentence boundaries within a window, as shown in Figure 2.1 [27]. P_k's error calculation only considers whether a chapter exists in a given window, while WD requires the number of chapters to be equal in a given window between the prediction and ground-truth in order for it not to be considered an error. Similar to accuracy, predicting all labels as negative still yields a decent score with both P_k and WD due to the low number of boundaries relative to the total number of sentences. Also, using a fixed window size k for the entire video assumes that the distance between boundaries has a uniform distribution, which might not necessarily be the case. For example, a video might have several small boundaries in the beginning but few boundaries towards the end.

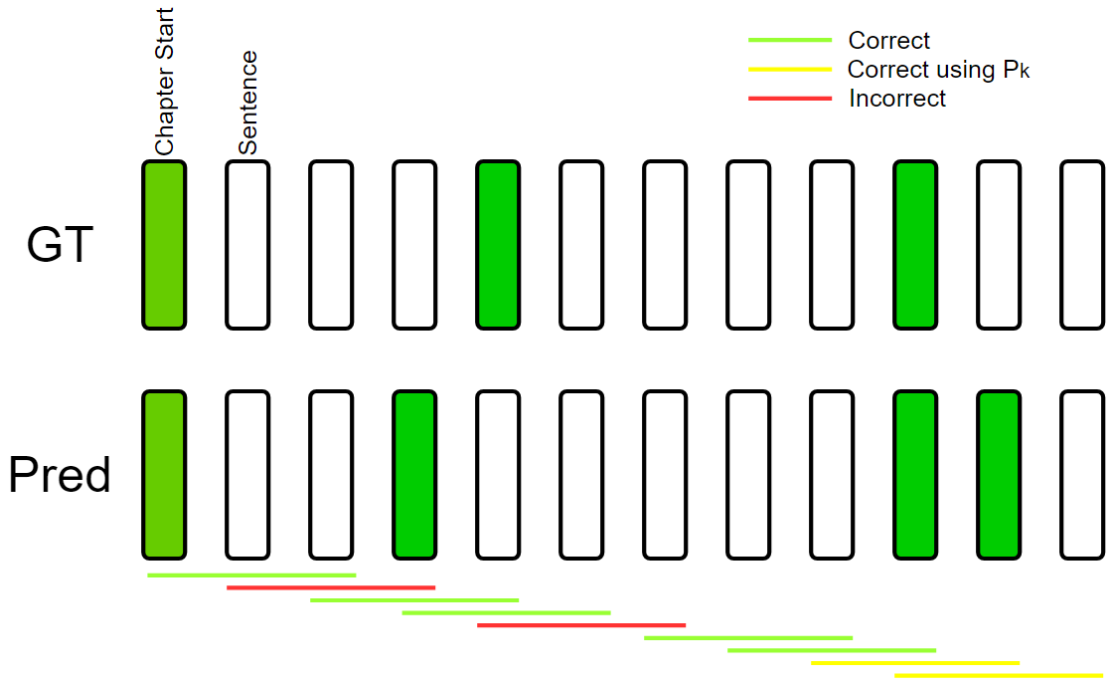


Figure 2.1: A visual illustration of the sliding window evaluation for the P_k and WD metric. The boxes represent sentences, with green representing sentences that are at the start of chapters for both ground truth (top) and prediction (bottom). The horizontal bars represent the sliding window for P_k and WD evaluation. P_k considers the yellow and red windows to be true positive predictions, while WD only considers the red windows to be true positive predictions.

2.2.2 *MIoU* and *EIoU*

In the temporal video segmentation task, Mean Intersection over Union ($MIoU$) is a commonly used metric [28]. Instead of comparing boundaries, segments formed from boundaries are compared. The segment lengths are based on time instead of sentence indices. It's formulated as:

$$MIoU = \sum_{i=1}^{m} \frac{1}{m} \max_{j} IoU(p_i, y_j) \tag{2.4}$$

where p_{i} is the ith predicted segment, g_{j} is the jth ground-truth segment. IoU is the

intersection over union between a predicted segment and the best ground segment match. The $MIoU$ metric has a few limitations, as specified by Ji et al. [20]. It only consists of the IoU of predicted matches, thus it only evaluates how precise the matched predicted segments are. It does not take into account superfluous predictions (i.e. extra predictions that don't match any ground truths) or missed predictions.

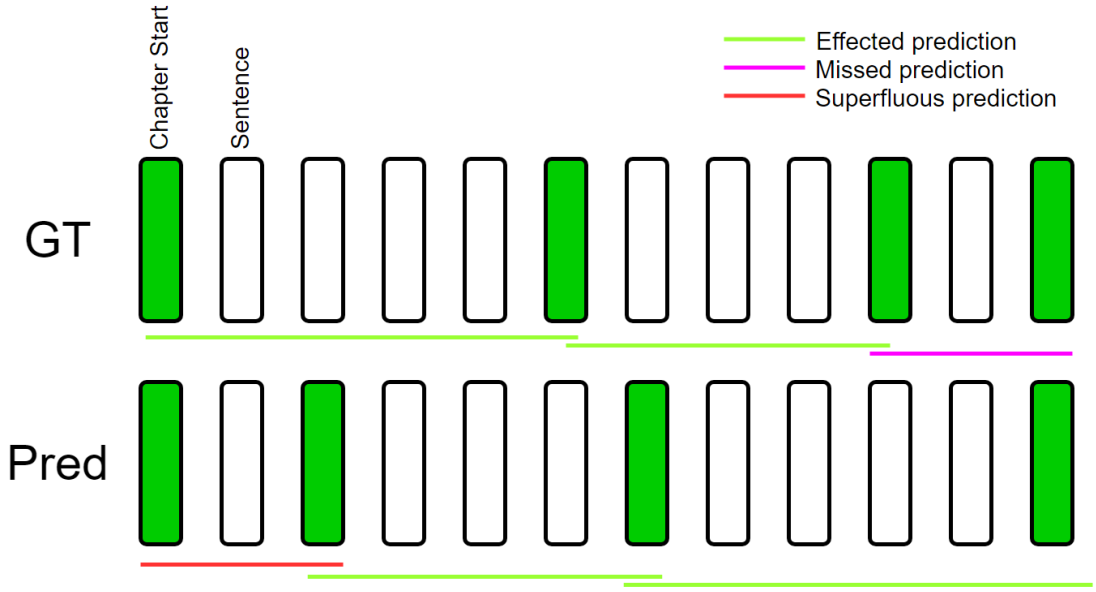


Figure 2.2: A visual illustration of superfluous prediction, effected prediction, and missed prediction.

To address this, Ji et al. [20] introduced a new metric called the Experienced Intersection Over Union ($EIoU$) which is evaluated as follows:

$$EIoU = \frac{n}{n + n_s + n_m} \sum_{i=1}^{n_e} \frac{L(p_i)}{L} \max_j IoU(p_i, y_j) \tag{2.5}$$

where $L(p_i)$ is the duration of the predicted segment p_i, L is the total duration of the video, n is the number of ground-truth segments, n_e is the number of effected predictions, n_s is the number of superfluous predictions and n_m is the number of missed predictions.

2.3 Techniques

2.3.1 Literature

Most recent techniques involve segmenting the transcript into sentence embeddings and passing the embeddings into a supervised learning model, shown in Figure 2.3. Koshorek et al. [21] first introduced this approach which involved embedding each word in a sentence using word2vec and then passing it into a Long Short-Term Memory (LSTM) network to obtain sentence embeddings which are then passed into another LSTM network for prediction. Barrow et al. [2] iterated on this approach but instead used FastText for embeddings, an improved version of word2vec. With the advent of transformers being a huge breakthrough for large language models (LLMs), Xing et al. [37] utilized the Bidirectional Encoder Representations from Transformers (BERT) model for sentence embeddings.

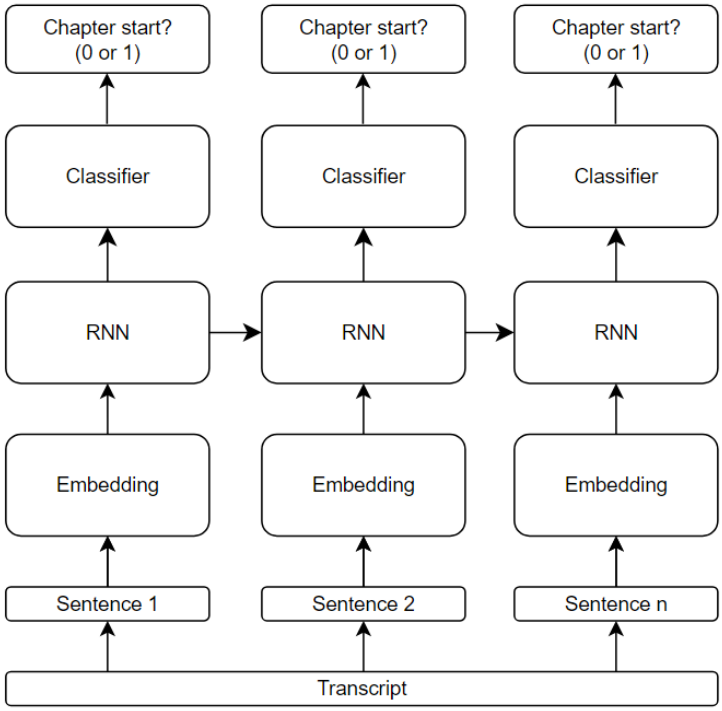


Figure 2.3: General formulation of text segmentation as a supervised learning task.

Alternatively, instead of taking into account the full context of every sentence in the text when making the inference, some techniques only take into account the local context

of nearby sentences, which can be represented as in Figure 2.4. Badjatiya et al. [1] consider only the local context of nearby sentences and use convolutional neural networks (CNN) for sentence embeddings and pass the local sentence embeddings into an LSTM. Lukasik et al. [23] introduced the cross-segment BERT approach, which encodes the local surrounding sentences around the target sentence as one segment using BERT. Lukasik et al. [23] also presented a hierarchical approach where sentences are encoded with BERT which are then together passed into another BERT model responsible for relating the sentences.

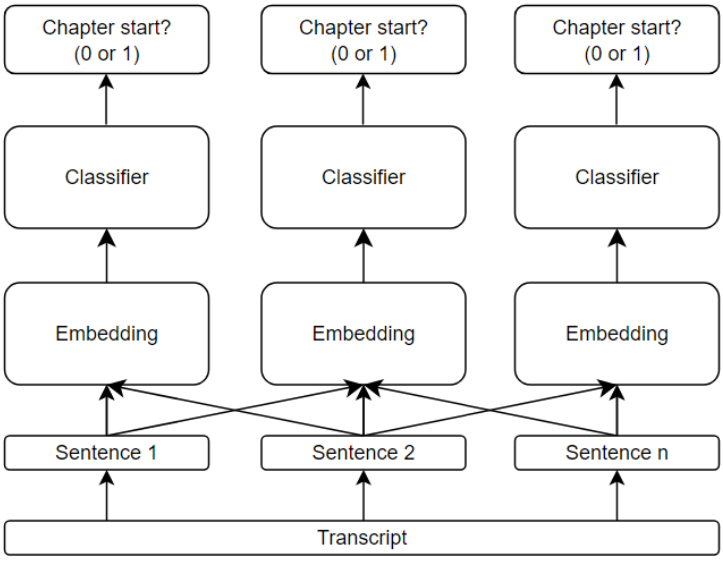


Figure 2.4: Alternate formulation of text segmentation as a supervised learning task. Instead of taking into account the full context as shown in Figure 2.3, this takes into account only the local context of nearby sentences by including nearby sentences in the embedding.

Although not as prevalent in recent research, there have been notable unsupervised methods performed. TextTiling is a classical approach that determines chapters based on similarity scores of word frequencies in nearby sentences [15], represented in Figure 2.5. Solbiati et al. [32] take the classical approach of TextTiling but instead utilize Sentence-BERT (SBERT) embeddings to compare similarity to. Other older unsupervised methods include GraphSeg [12] and BayesSeg [12]. The advantage to an unsupervised approach is there is no need for training and labelled data, which may or may not be readily available depending on the domain.

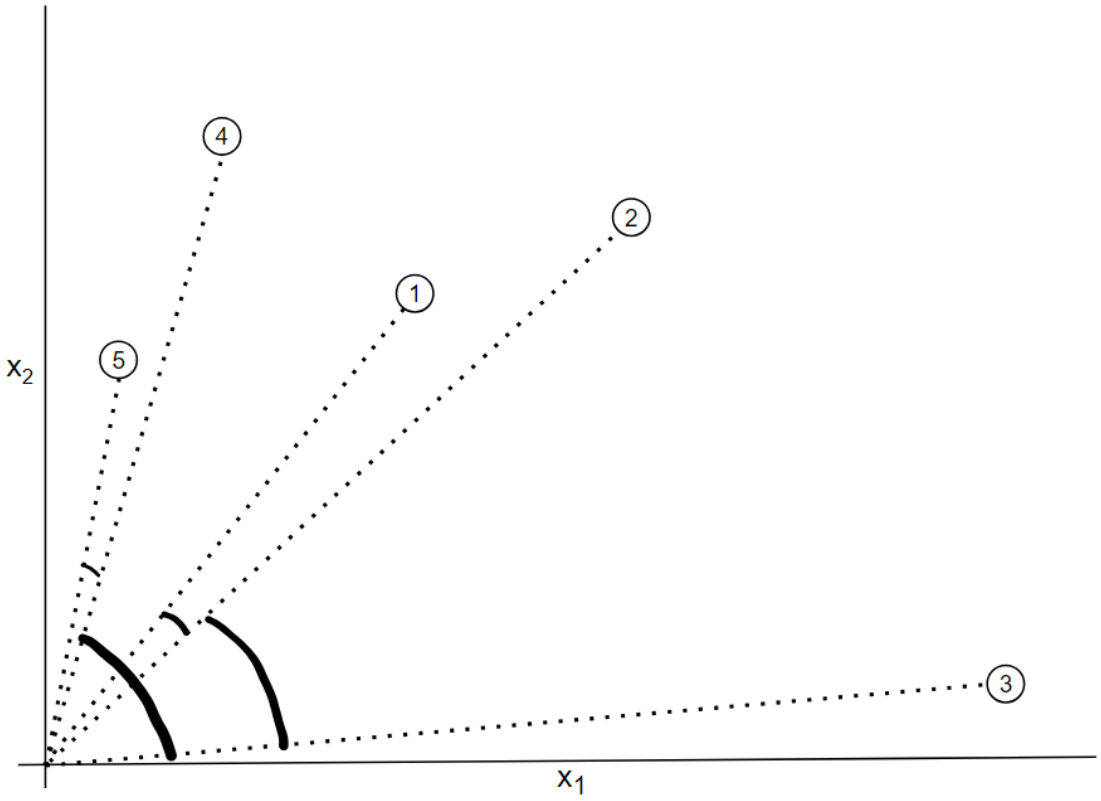


Figure 2.5: Example of a cosine similarity comparison for an unsupervised approach to topic segmentation. The cosine between two feature vectors from neighbouring sentences can be compared and thresholded to determine if the two feature vectors are different enough that it would be considered a new topic.

2.3.2 Commercial

Recently, YouTube has begun including automated chaptering in videos as a feature that video creators can enable if there are no manual chapters already present in the video [16]. According to YouTube, they use machine learning on videos to recognize text (i.e. optical character recognition) to inform the automated chaptering [31]. This seems partially correct, as indeed there are some examples of videos, as shown in Figure 2.6, which seem to provide automated chapters based on the text content of the video, especially when the video is structured in a consistent way like a step-by-step instructional video. However, there are videos with no text content in the video that have received automated chapters,

also shown in Figure 2.6, suggesting YouTube also is able to chapter videos based on the auto-generated transcript.

Sometimes, instead of chapters, the video may have "key moments" instead [6]. Key moments, like chapters, can be both auto-generated and user-created, although it is harder to distinguish when the user has set their own key moments versus when YouTube auto-generated them. It is worth noting that not all videos are eligible for automated key moments or chaptering, many videos exist on YouTube without any chapters despite the creator enabling auto-generated chapters [30]. This suggests that YouTube isn't yet confident with their chaptering method to chapter every video. More importantly, it makes comparing how effective their chaptering implementation performs challenging. A good test for performance would be to upload an already user-chaptered video and compare it to what YouTube generates, but the requirements to get YouTube to chapter a video are vague.

Microsoft has recently introduced chaptering as a part of their intelligent recap for meetings done in Microsoft Teams [17]. Microsoft claims to leverage OpenAI's GPT for the intelligent recap [17]. It only works for meetings in Microsoft Teams, so existing meeting datasets can't be easily used for comparison. However, meetings can be chaptered using OpenAI's ChatGPT. ChatGPT is a chatbot built on their generative pre-trained transformer (GPT) models, capable of performing many tasks when prompted by a user with natural language [35]. It consists of billions of parameters pre-trained from text on the internet and is fine-tuned through extensive Reinforcement Learning from Human Feedback (RLHF) to be able to predict the answer given natural language [35].

Other similar available generative large language models like Llama 2 and Mistral do not seem to interpret the chaptering task when queried, they respond by summarizing the transcript instead.

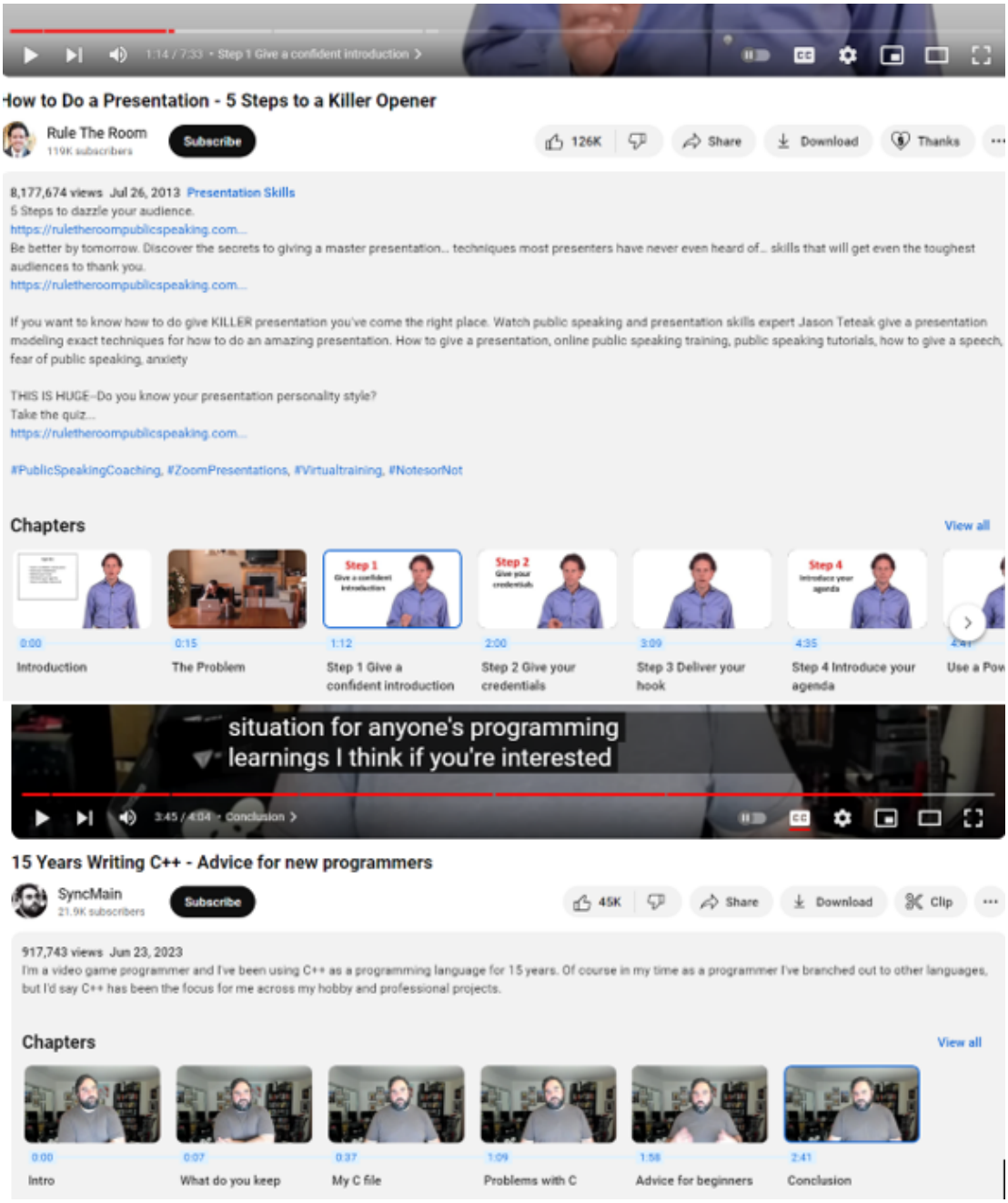


Figure 2.6: Examples of videos automatically chaptered by YouTube. The top video is likely chaptered based on text in the video, while the bottom video is likely chaptered based on the transcript.

Chapter 3

Experimental Setup

3.1 Datasets

The AMI corpus will be the primary dataset used for evaluation, as it contains a sufficient number of meetings for testing. Furthermore, the AMI corpus provides a strongly encouraged train, validation, and test split which makes the comparison to other methods more consistent [5]. The AMI corpus does not have a large amount of videos available to be used for training, however. Transfer learning, which involves training a model on a large dataset of a different domain and then fine-tuning it on the target domain with a smaller dataset, has been very beneficial to many solutions involving deep learning [18]. The ICSI, Wiki-727k and a custom-curated YouTube dataset will be used to investigate the effectiveness pre-training can provide in the chaptering task. ICSI is used as it is the most similar to AMI in terms of content as well as granularity of chapters, due to both datasets having hierarchical topic segment annotations. Wiki-727k was chosen as it contains the largest amount of samples of any dataset for topic segmentation, while also having a high granularity of chapters due to it containing hierarchical sections. Details of the custom YouTube dataset are discussed in Section 3.1.2. All datasets used for pre-training, besides AMI, are randomly split 80-10-10 for training, validation, and testing.

3.1.1 Pre-Processing

Both AMI and ICSI provide the chapters as well as individual words spoken in timestamps, thus some pre-processing needs to be done to prepare the input to be suitable for topic segmentation. Figure 3.1 shows a snippet of raw meeting data from AMI.

Words

@who	@punc	@trunc	@mispronounced	@pron	start()	end()	text()
A					927.94	929.23	Okay
A					929.23	929.51	then
A					929.51	929.67	we
A					929.67	929.9	have
A					929.9	930.28	fifteen
A					930.28	930.47	point
A					930.47	930.59	eight
A					930.59	930.96	Euros
A	TRUE				930.96	930.96	
A					932.93	933.1	It
A					933.1	933.41	wasn't
A					933.41	933.76	bad
A	TRUE				933.76	933.76	
A					937.17	937.54	Okay
A					937.54	938.13	so
A					939.24	939.44	we're
A					939.44	939.89	on
A					939.89	940.29	to
A					940.29	940.61	the
A					942.96	943.08	We're
A					943.08	943.21	on
A					943.21	943.39	to

Topics

@other_description	start()	end()
	3.57	74.302
	75.48	435.216
	120.74	340.01
	340.01	374.08
	437.24	777.11
	779.57	942.78
	942.96	1224.89
	1224.89	1250.19

Figure 3.1: Raw sample of AMI meeting IS1002d. The start and end timestamps for the utterances of each word are provided along with the speaker, along with if the utterance is considered punctuation. Similarly, the start and end timestamps for each topic are provided. The topics that start before another ends are considered to be sub-topics, an example of which can be seen in the two topics that follow the second topic.

Words are combined into sentences by grouping the words until a punctuation that is not a comma is encountered. Filler words such as "umm" are discarded, as well as sentences that are less than four words. The sentence during or the immediate sentence following the start of a topic timestamp is considered to be the starting sentence of the topic. If two individuals speak at the same time (i.e. the start and end timestamps of sentences overlap), the order is determined by the start timestamp of the sentence. Figure 3.2 shows a snippet of processed meeting data for topic segmentation originating from Figure 3.1.

3.1.2 Custom YouTube Dataset

In 2020, YouTube launched a feature where video creators can manually chapter their videos by providing the timestamps to the chapters in the description, as shown in Figure

Speaker	Sentence	Label
B	It was a pretty accurate estimate I would say.	0
A	Okay so we're on to the We're on to the pat-on-the-back part of the presentation, where we have a look at the criterias that Paul the criterion criteria that Paul has has given us, and we can use that to tell How's it going?	1
B	s kind of frighteningly accurate.	0
A	Anyone got any thoughts?	0
A	How how have we done today?	0
B	I think we did pretty well.	0

Figure 3.2: Pre-processed sample of AMI meeting IS1002d for topic segmentation. Speaker B's second sentence was uttered during Speaker A's first sentence. This sample shows an edge case with pre-processing as Speaker B's second sentence is a continuation of their first sentence yet is appearing after the topic boundary.

3.3 [16]. Since then, there has been a huge quantity of videos chaptered by creators which is ripe to form a large dataset sufficient for training. YouTube videos, which contain spoken monologue and tend to be more informal, are theoretically closer to the domain of meetings than Wikipedia articles or other written text.

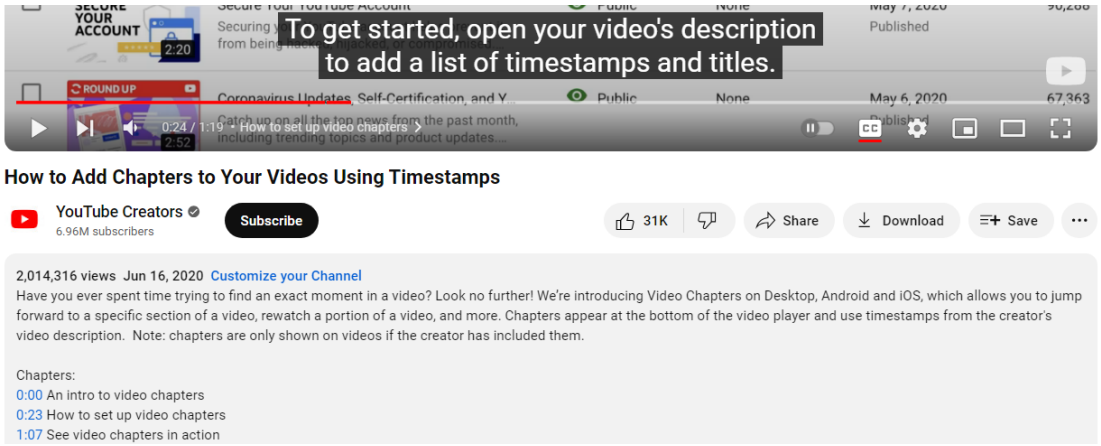


Figure 3.3: Example of a manually chaptered YouTube video.

Two sets of tutorial videos are obtained, named YT-Programming and YT-Design. Tutorial videos are retrieved because they are easy to automatically collect from a search

query (using pytube) and are the types of videos that are very likely to be chaptered by the creator. YT-Programming contains programming tutorial videos obtained by searching for videos with the following keywords:

- programming tutorial
- coding tutorial
- cpp tutorial
- java tutorial
- javascript tutorial
- python tutorial
- kotlin tutorial

YT-Design contains design tutorial videos obtained by searching for videos with the following keywords:

- adobe tutorial
- after effects tutorial
- blender tutorial
- gimp tutorial
- illustrator tutorial
- maya tutorial
- photoshop tutorial

Only videos in English that have a list of manual chapters in the description of the video are collected, following YouTube's manual chapter format guidelines [16]. Timestamps to the start of the chapters are retrieved from the description.

Not all videos will have transcripts available, and auto-generated captions from YouTube do not contain punctuation, as seen in Figure 2.6, which would make it difficult to prepare for topic segmentation. Thus, for consistency, all videos obtained are processed through Microsoft Stream to obtain auto-generated transcripts in VTT format which do contain punctuation and thus can be divided into sentences. Similar to how AMI and ICSI were pre-processed in Section 3.1.1, the sentence during or the immediate sentence following the start of a chapter timestamp is considered to be the starting sentence of the chapter.

3.1.3 Summary and Statistics of Datasets

The statistics for each dataset used are in Section 3.1. For the datasets that have chapter hierarchies, hierarchies are flattened for training and evaluation, which gives a high granularity of chapters. Many topic segmentation research uses only the top-level segmentation and discards the sub-chapters within chapters [21] [32]. Ultimately, the choice of whether to include sub-chapters falls under user preference, which varies from person-to-person and domain. Training with a high granularity of chapters should produce a model that outputs a high granularity of chapters. Also, chapters and sub-chapters themselves can have

different interpretations across domains, which is exemplified by how the different domains vary in length between sub-chapters. ICSI meetings tend to be longer than AMI meetings but contain a similar number of sentences per chapter on average. Wikipedia articles are on average much shorter and as well as has shorter chapters. YT-Programming videos are much longer and contain longer chapters, though with a large variance. YT-Design videos have closer statistics to AMI meetings than YT-Programming, despite YouTube videos not containing any sub-chapters.

Dataset	# Samples	Avg # Sentences in Samples	Avg # Chapters in Samples	Avg # Sentences in Chapters	Avg # Words in Sentences
AMI Meetings	135	279 ± 129	12 ± 7	22 ± 29	14 ± 12
ICSI Meetings	75	520 ± 185	19 ± 10	26 ± 27	17 ± 14
Wiki-727K	727746	48 ± 59	4 ± 5	9 ± 9	21 ± 12
YT-Programming	660	854 ± 1370	11 ± 14	66 ± 113	14 ± 11
YT-Design	937	261 ± 423	7 ± 8	31 ± 46	15 ± 12

Table 3.1: Summary and statistics of datasets.

3.2 Model Architectures

4 architectures adapted from existing literature are used and investigated. All supervised learning approaches take the best performing model, with respect to WD and $EIoU$, through 200+ epochs on the validation set. Each model was trained with PyTorch using the AdamW optimizer with a .0005 learning rate. The GPU used was the NVIDIA GeForce RTX 2080 Ti.

3.2.1 SBERT LSTM

The first model, inspired by Xing et al. [37], consists of acquiring embeddings for each sentence using a pre-trained Sentence-BERT (SBERT) model and passing each sentence to a Bidirectional-LSTM. SBERT is a modification of BERT designed to derive meaningful sentence embeddings through the use of Siamese and triplet network structures [29]. The

pre-trained 'all-MiniLM-L12-v2' SBERT model included in the Sentence-BERT Python framework [29] encodes a sentence into a 384-length vector and is frozen during training. This model learns the relationship of topic segmentation across sentences. The structure is shown in Figure 3.4.

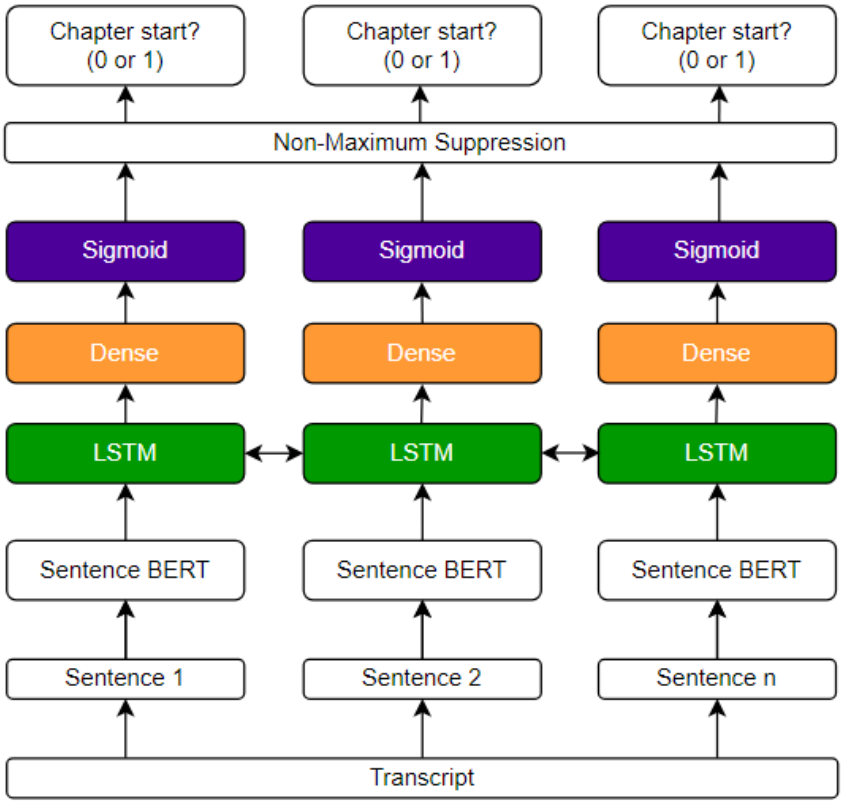


Figure 3.4: The SBERT LSTM model for chaptering. The Bi-LSTM consists of 2 layers, with a dropout of 0.5 leading to a dense layer that encodes a 256-length vector into a 1-length vector.

In total, it contains 921 857 trainable parameters and 33 360 000 frozen parameters in the 'all-MiniLM-L12-v2' SBERT model. The SBERT embeddings of each sentence in the training data can be cached after being computed once to speed up training time. A batch size of 5 was used for training.

3.2.2 Cross-Segment BERT

The second model Cross-Segment BERT (CS BERT), adapted from Lukasik et al. [23], consists of passing tokens before and after the start of a sentence to a pre-trained BERT [7] model with a single fully connected layer classification head. This model assumes that the local context is sufficient information to chapter a video. The structure is shown in Figure 3.5.

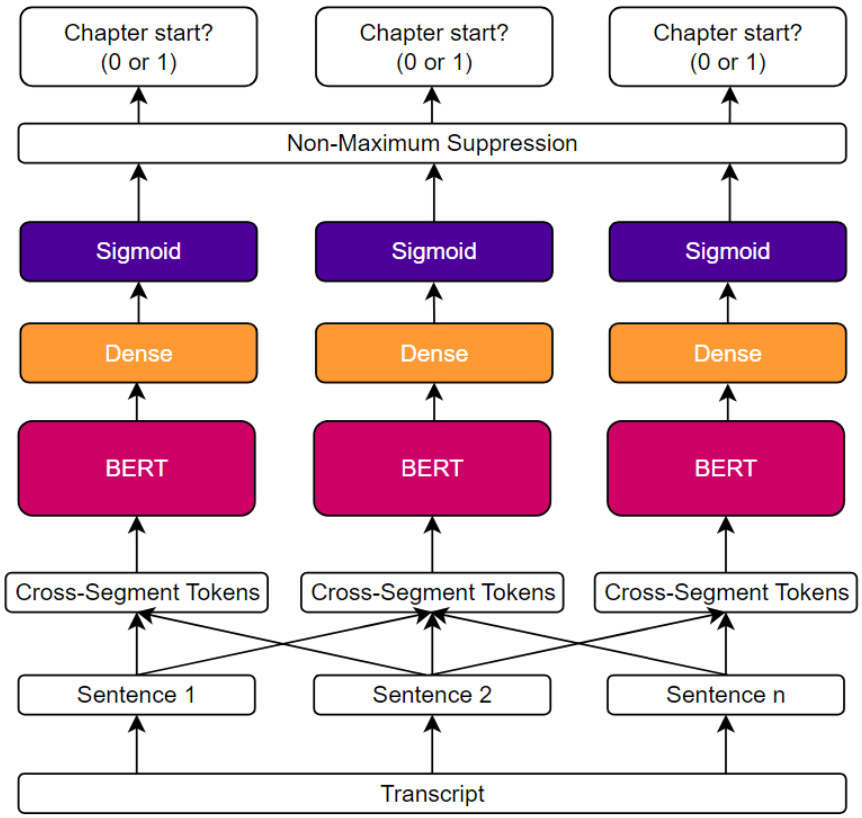


Figure 3.5: The Cross-Segment BERT model for chaptering. A max of 100 tokens or 5 sentences tokenized by the BertTokenizer are passed to the 'bert-base-uncased' model implemented by the transformers Python framework, which encodes the tokens into a 768-length vector. Each set of cross-segment tokens starts with a classification token [CLS] and sentences to the right and left of the target sentence are separated by a separator token [SEP].

The BERT model can be fine-tuned by gradually unfreezing layers. First, the BERT model is frozen and only the classification head is trained, and then the last two layers of BERT is gradually unfrozen. In total, the model contains 109 483 778 total parameters, with 14 177 282 of those parameters being trainable. Due to memory limitations, a batch size of 1 is used and the max number of sentences processed in a single batch is 500. If the number of sentences in the input exceeds 500, a random 500-length sentence window is acquired from the input.

3.2.3 Cross-Segment BERT LSTM

The third model, Cross-Segment BERT LSTM (CS BERT LSTM) combines the previous two approaches in Section 3.2.1 and Section 3.2.2 and connects each segment's BERT embeddings to a Bi-LSTM. This approach, inspired by Lukasik et al. [23], is hierarchical in that the BERT model is responsible for learning the relationship within a local segment while the Bi-LSTM is responsible for learning the relationship across segments. The structure is shown in Figure 3.6.

In total, it contains 110 797 570 trainable parameters and 15 491 074 of those parameters being trainable. The same memory considerations as in Section 3.2.2 applies: a batch size of 1 is used, the maximum number of sentences processed in a batch is 500, and a random 500-length sentence window is acquired from the input if the number of sentences is greater than 500.

3.2.4 SBERT Text-Tiling

SBERT Text-Tiling, introduced by Solbiati et al. [32], is an unsupervised method that detects topic changes based on similarity scores between sentences. It is done by computing obtaining SBERT embeddings for every sentence, max-pooling adjacent embeddings into blocks, and then computing the cosine similarity, sim_i, between each adjacent block. The similarities between each block are then smoothed using a uniform sliding window and then thresholded by:

$$T = \{i \in [0, M] | sim_i < t * max(sim)\} \tag{3.1}$$

where $max(sim)$ is the max similarity between adjacent blocks in the transcript, and t is a threshold constant. Figure 3.7 shows a diagram illustrating this approach. The same parameters as Solbiati et al. [32] are used which sets t as 0.6, the number of sentences in a max-pooling block as 15, and the smoothing window between blocks as 1.

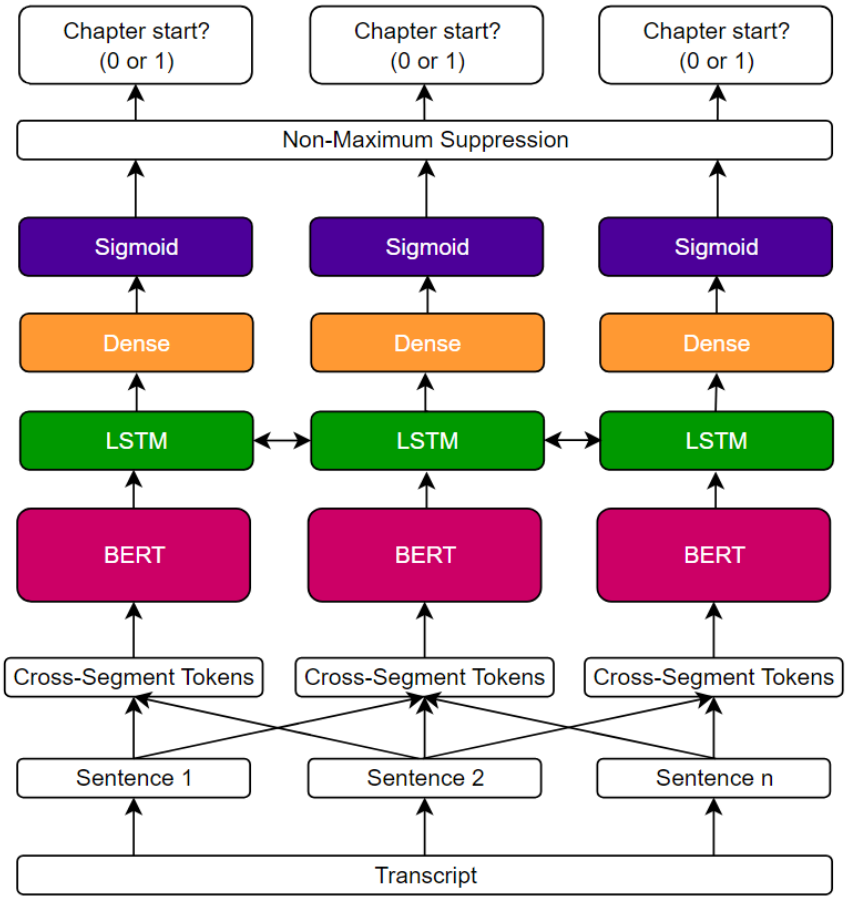


Figure 3.6: The Cross-Segment BERT LSTM model for chaptering.

3.3 Loss Function

A standard weighted binary cross entropy (BCE) with logits loss is employed, which can be represented as:

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}[w_p y_i \cdot \log \sigma(x_i) + (1 - y_i) \cdot \log(1 - \sigma(x_i))] \tag{3.2}$$

where x_n is a sentence, $\sigma()$ is the model, y_n is the ground-truth label, w_p is the weight for the positive sample, and n is the number of sentences.

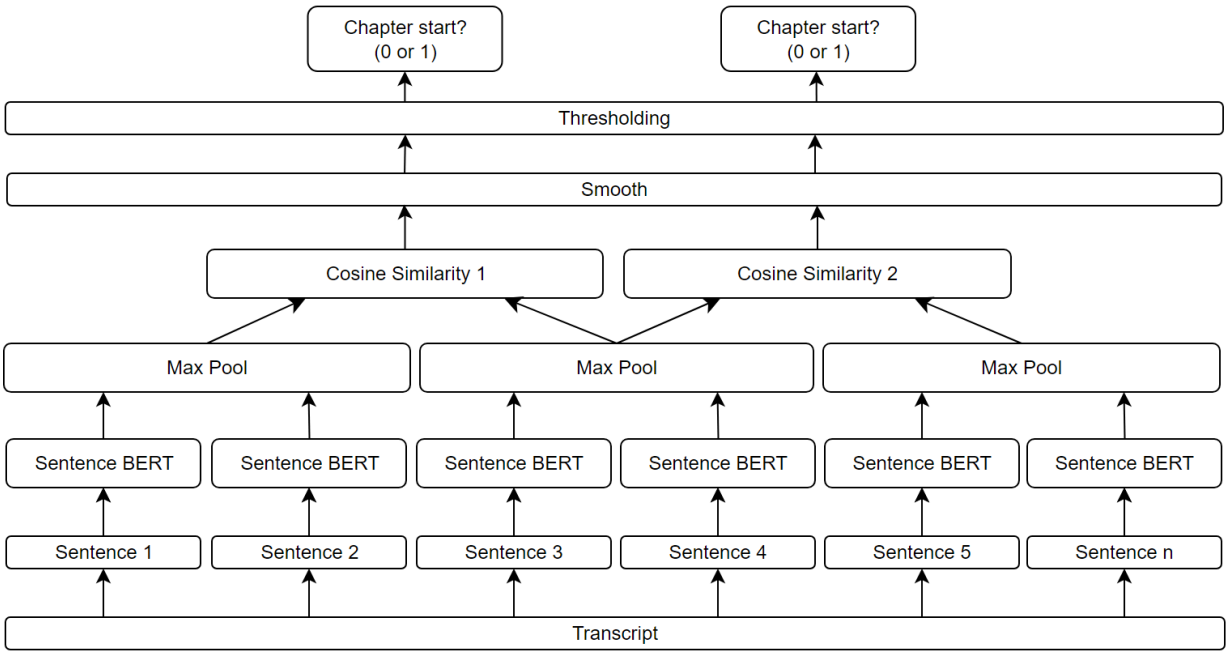


Figure 3.7: SBERT Text-Tiling approach. The number of sentences included in the max pool window can be adjusted.

Naturally, there will be far fewer sentences that are starts of chapters than not, so the positive class is given a weight calculated by the ratio of the number of negative samples to the number of positive samples. Focal loss can also be used to address class imbalance, as well as prevent the model from only classifying easy examples well [22]. Focal loss is represented as:

$$L_{Focal} = -\frac{1}{n}\sum_{i=1}^{n} -[(1-\alpha)(1-\sigma(x_i))^{\gamma} y_i \log \sigma(x_i) + \alpha\sigma(x_i)^{\gamma}(1-y_i)\log(1-\sigma(x_i))] \quad (3.3)$$

where γ is the focusing parameter which makes confident or 'easy' matches down-weighted more than less-confident or 'harder' matches and α is the weighting factor based on class distribution. γ is set as 2 as it got the best results when it was introduced by Lin et al., while α is calculated by the frequency of positive class samples in a batch [22].

3.4 Label Smoothing

A difficult nature of this problem is the nature of the ground-truth is subjective. Individuals may have different perspectives in segmenting a video. The sample excerpt of a meeting transcript in Figure 3.8 highlights this. Furthermore, there is potential for noise, as shown in Figure 3.2.

Speaker	Sentence	Label	Smooth
A	Okay then we have fifteen point eight Euros.	0	0
C	It's a really crazy dog Okay.	0	0.0498
C	It's a dog in a village.	0	0.1353
C	So what are you going to do now?	0	0.3679
A	Just have to present project, discuss a little bit about it.	1	1
D	, we have twenty five minutes for the meeting.	0	0.3679
A	So, if you have questions.	0	0.1353
D	Know what time is it?	0	0.0498

Figure 3.8: Pre-processed sample of AMI meeting IS1003a for topic segmentation. This shows how the exact start of a topic is open to subjectivity. The sentence before the labelled topic boundary could have been chosen as the starting sentence. Smoothing can be employed to the labels around the true boundary.

The nature of the noisy and subjective labels can cause potential valid starts of sections to be penalized by the loss function. Label smoothing can be employed so that sentences near ground-truth labels are much less penalized. This also has the effect of distributing more positive samples, since the nature of the problem makes the training set heavily imbalanced toward negative samples. A Gaussian-like blur can be applied to the training labels, as shown in the following equation:

$$y_n = \begin{cases} e^{\frac{-d_{nearest}}{k}} \; if \; d_{nearest} <= cutoff \\ 0 \; if \; d_{nearest} > cutoff \end{cases} \tag{3.4}$$

where $d_{nearest}$ is the temporal distance in sentence indices to the closest positive boundary. The cut-off was arbitrarily chosen to be a length of 3 sentences.

3.5 Augmentation

Data augmentation can help artificially increase the size and diversity of the dataset by creating more training data from transforming existing training data. Small datasets are prone to overfitting, so increasing the dataset through augmentation can help remedy that.

The augmentation technique employed is generating altered transcripts by taking snippet windows of existing transcripts and randomly picking sentences to exclude. A visualization of this windowing augmentation method is shown in Figure 3.9. For every transcript in a batch, 3 to 6 window snippets are generated. The position of the window and window length is picked randomly, with the window length having a length between 2 and 100. Each sentence has a dropout chance of 25%. The dropout also helps mimic the subjectivity and noise of ground-truth labels, as if the sentence that is normally the chapter start is dropped, the sentence after becomes the chapter start. It may also help the model not rely on transition sentences such as "Okay, now we're..." to chapter, which are common in many but not all chapter starts.

3.6 Ensemble

As mentioned previously, the chaptering task is subjective. A transcript could be segmented in many different ways by different people. Ground-truth text segmentation for the ICSI corpus had multiple annotators responsible for chaptering, and the inter-annotator agreement was evaluated [14]. A similar approach can be mimicked with classification models. Instead of using a single trained model to classify, multiple trained models can be used for classification. Multiple datasets are already being used to train the same type of model to analyse the effectiveness of transfer learning. The output of these models trained on different datasets can be combined through averaging, resembling a common ensemble approach called bagging [8]. This approach is represented visually in Figure 3.10.

A max pool is employed with a kernel size of 9 and stride of 1 so that close indices with positive labels in each model's output can be averaged together as shown in Figure 3.11.

3.7 ChatGPT

The effectiveness of the generative large language model ChatGPT on the chaptering task will be evaluated. The following query is sent to the latest version of ChatGPT 3.5 as of Feb 13, 2024:

```
My name is  Danny Wolfs.
 this is the agenda for today.
 first a little opening.
 I will introduce myself,  and  I think it's very  good to introduce  yourself.
 then  a little bit of acquaintance, acquaintance to  to to ourselves.

 that will be done  with a tool training from the these two  smart boards.
Then the project plan.
What we're going to do, and how we're going to do it.
============Chapter 0============

Okay , my name is  Danny Wolfs.
I'm the Project Manager.
I'm the User Interface Designer.
User interface, okay.

 my function is the Marketing Expert.
============Chapter 1============
First a little about the project aim.
 the the the aim is to make a new remote control.
 maybe you have read  read the website.
It's a very , yeah, very  ambitious  company.


 first of all  it must be original,  and trendy.

But at the same time  user-friendly.
And they have  Yeah, that's  very important  for them.
 there are three stages.
There is a functional design.
```

Figure 3.9: A sample augmentation. A random snippet of the transcript is obtained, with some sentences dropped out.

```
Please chapter the following meeting transcript and provide the sentence at the
    start of each chapter.
```

followed by a copy and paste of the transcript, with each sentence separated per line. ChatGPT will respond with an output resembling Figure 3.12. The starting sentences outputted by ChatGPT are then manually converted into a vector containing the index of the chapters to be used to compare with ground truth. ChatGPT will sometimes chapter the transcript out of order. When that happens, the response is regenerated until the chapters are given in the right order.

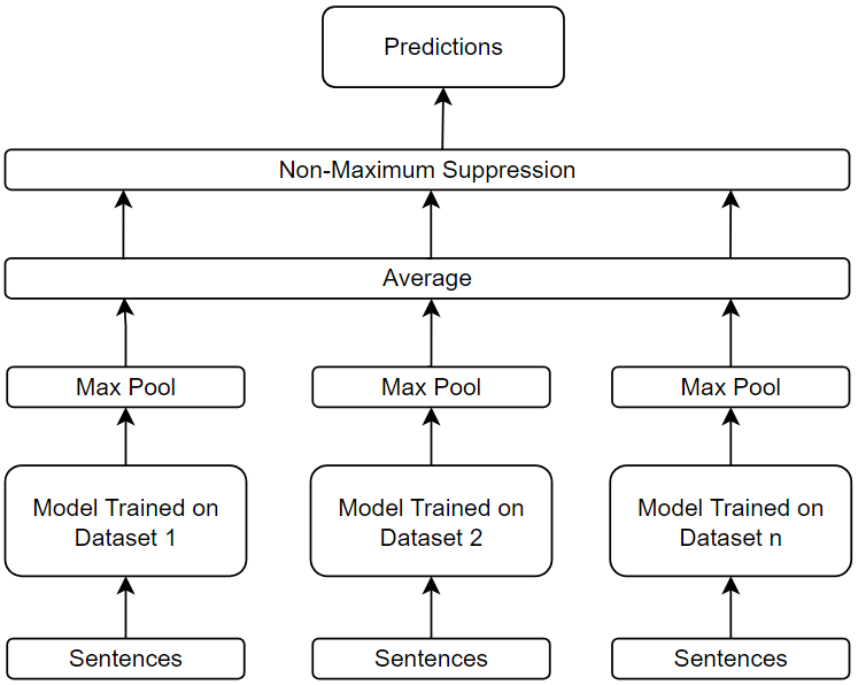


Figure 3.10: Ensemble bagging methodology.

3.8 Summary

With all these techniques in mind, the next chapter will compare their effectiveness for the meeting chaptering task. Meeting chaptering is challenging due to the subjectivity of ground-truth labels, as detailed in Section 3.4. Label smoothing and an ensemble approach are techniques that in theory could help with noisy labels, while label smoothing and augmentation could help reduce overfitting the small training data available in the meeting domain. Furthermore, with meetings being a different domain than written text, as discussed in Section 3.1, it will be observed whether pre-training on established large written text datasets or the newly curated YouTube dataset is effective.

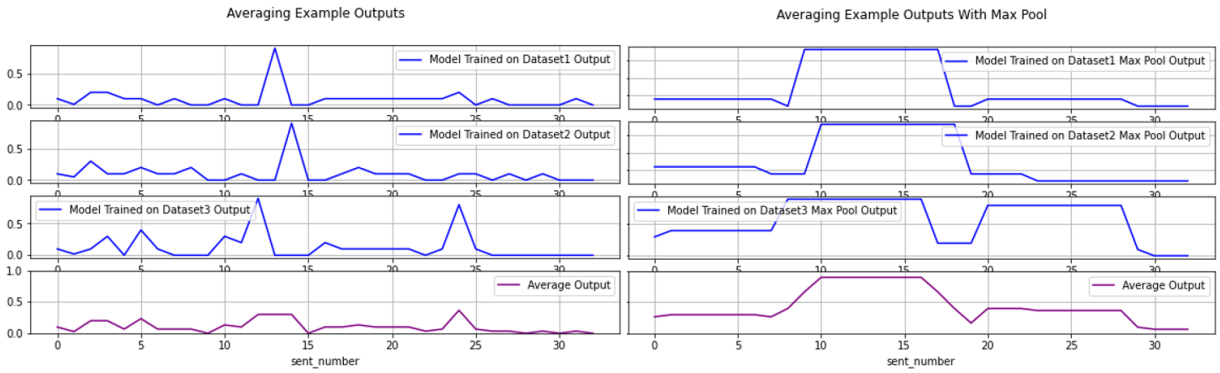


Figure 3.11: Sample ensemble bagging example. This example shows how averaging outputs of models can help improve precision. The top three charts show example outputs from three different models and the bottom chart shows the average output, with the charts on the right showing the model's output with a max pool applied. The third model's output predicts a second potential chapter start, but because the other models do not predict that sentence as being likely to be a chapter start, the average output would not predict it as a chapter. The majority of models have to have high confidence that the sentence is a chapter start. Furthermore, this example also shows how max pooling the outputs before averaging can be beneficial. Each model may predict different but close-together sentences as the start of the chapter. Without max pooling, these would get averaged so that no sentence is considered a chapter. Some leniency is desired, as labelling the exact sentence as the start of the chapter correctly is less of a concern than missing the chapter start entirely.

You

Please chapter the following meeting transcript and provide the sentence at the start of each chapter:

I have some new findings on Marketing Expert level, which I will show you.
The method I used was giving orders to our usability lab to do a questionnaire.
one hundred respondents were involved and my marketing department generated a report with a lot of results.
, these were a couple of findings, first page of three.
, we have three audiences of two audiences, I'm sorry.
the first one, this scale, from sixteen to forty five age.

ChatGPT

Chapter 1: New Findings on Marketing Expert Level

"I have some new findings on Marketing Expert level, which I will show you."

Chapter 2: Audience Segmentation and Preferences

"These were a couple of findings, first page of three. We have three audiences of two audiences, I'm sorry."

Figure 3.12: A sample query to ChatGPT.

Chapter 4

Observations

This chapter reveals and discusses the results of the experiments detailed in the previous chapter. First, a comparison of each model's performance on the AMI dataset is presented, followed by a comparison of utilizing different datasets for pre-training to show if pre-training is effective despite the domains being different. Next, the effectiveness of label smoothing, augmentation and an ensemble approach is observed. Finally, this chapter provides a critique of the metrics used for topic segmentation and re-contextualizes the results under a new proposed metric.

4.1 Effect of Different Models

Models	Trained On	$P_k \downarrow$	$WD \downarrow$	$MIoU \uparrow$	$EIoU \uparrow$
None	N/A	0.4050	0.4050	0.3643	0.1883
Even	N/A	0.4827	0.5311	0.4568	0.3523
Random	N/A	0.4662	0.5254	0.3785	0.3341
ChatGPT 3.5	N/A	0.3397	**0.4013**	0.4618	0.3978
SBERT Text-Tiling [32]	N/A	0.4328	0.4419	0.4754	0.3043
SBERT LSTM [37]	AMI	0.3596	0.4281	0.4547	0.3909
CS BERT [23]	AMI	**0.3265**	0.4031	0.4601	**0.3984**
CS BERT LSTM [23]	AMI	0.3454	0.4048	**0.5286**	0.3881

Table 4.1: Performance of different chaptering methods on AMI dataset.

Table 4.1 shows the results of each method on the AMI dataset. None, Even, and Random are used as naive baselines to compare chaptering results with the intuition that the chaptering method should at least be better than dividing the video evenly, randomly, or not at all to be effective. Even was implemented by dividing chapters evenly based on the length of the video and the number of chapters in the ground truth. Random was implemented by dividing the number of sentences by the number of chapters in the ground truth and using that as a probability that there is a chapter. None is labelling no sentence as a chapter.

The results indicate that P_k and WD penalize false positives as equally as false negatives, as labelling no sentences as chapters grants a decent score for both metrics. $MIoU$ and $EIoU$ favours false positives over false negatives, as even naive methods like random and even increase $MIoU$ and $EIoU$ considerably. This shows the value of having both sets of metrics to evaluate the different methods. The favourable metric is whether under-segmentation or over-segmentation is preferred by the user, though ideally methods should perform well using both metrics and be able to at least beat the naive methods on all metrics.

All methods generally perform better than the naive baselines with regard to each metric. The supervised methods generally perform better than the unsupervised text-tiling method. Of the supervised methods, the methods using CS BERT perform the best, which suggests that the local context of nearby sentences is adequate to chapter a video. ChatGPT performs competitively, it is very effective at chaptering out of the box without being tuned specifically for the chaptering task. The chaptering results of each method on sample AMI meeting TS3007a are displayed in Figure 4.1 as well as Appendix A where the chapter locations in the full transcript are shown.

Figure 4.2 which displays the training and validation curves for each metric shows that the supervised models overfit the training data considerably. The SBERT LSTM model perfectly chaptered the training data after around 200 epochs, while the performance for validation data lags behind. Figure 4.2 also shows that the P_k and WD metrics don't improve across epochs for the validation set, however $MIoU$ and $EIoU$ do. Again, this is because P_k and WD penalize false positives just as much as false negatives, so as the model starts to predict sentences as being chapters, the incorrect chapter predictions have equal weighting to the correct chapter predictions. If just observing the validation curves for P_k and WD it may seem like the model is failing to generalize anything, however observing the validation curves for $MIoU$ and $EIoU$ shows some generalization is likely occurring.

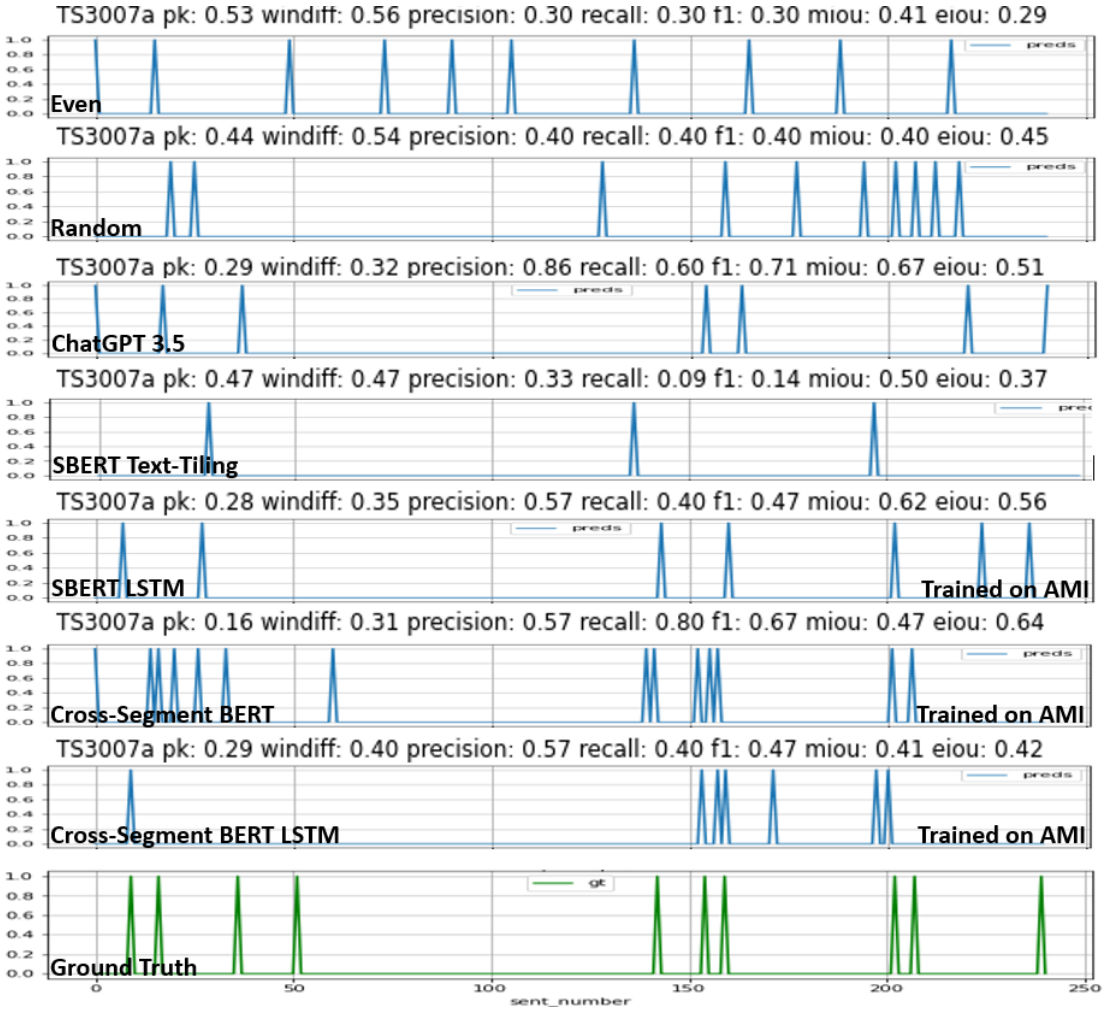


Figure 4.1: Predictions of each chaptering method on sample meeting TS3007a. All methods seem to effectively not label sentences as chapters in the middle of the transcript, showing some understanding that no new topics are being presented during that portion.

4.2 Effect of Transfer Learning

When training on Wiki-727k, SBERT LSTM does not overfit, as seen in Table 4.2 which shows that the model's performance on the training, validation and test sets are similar. This suggests that the overfitting is a data problem and that more videos should be

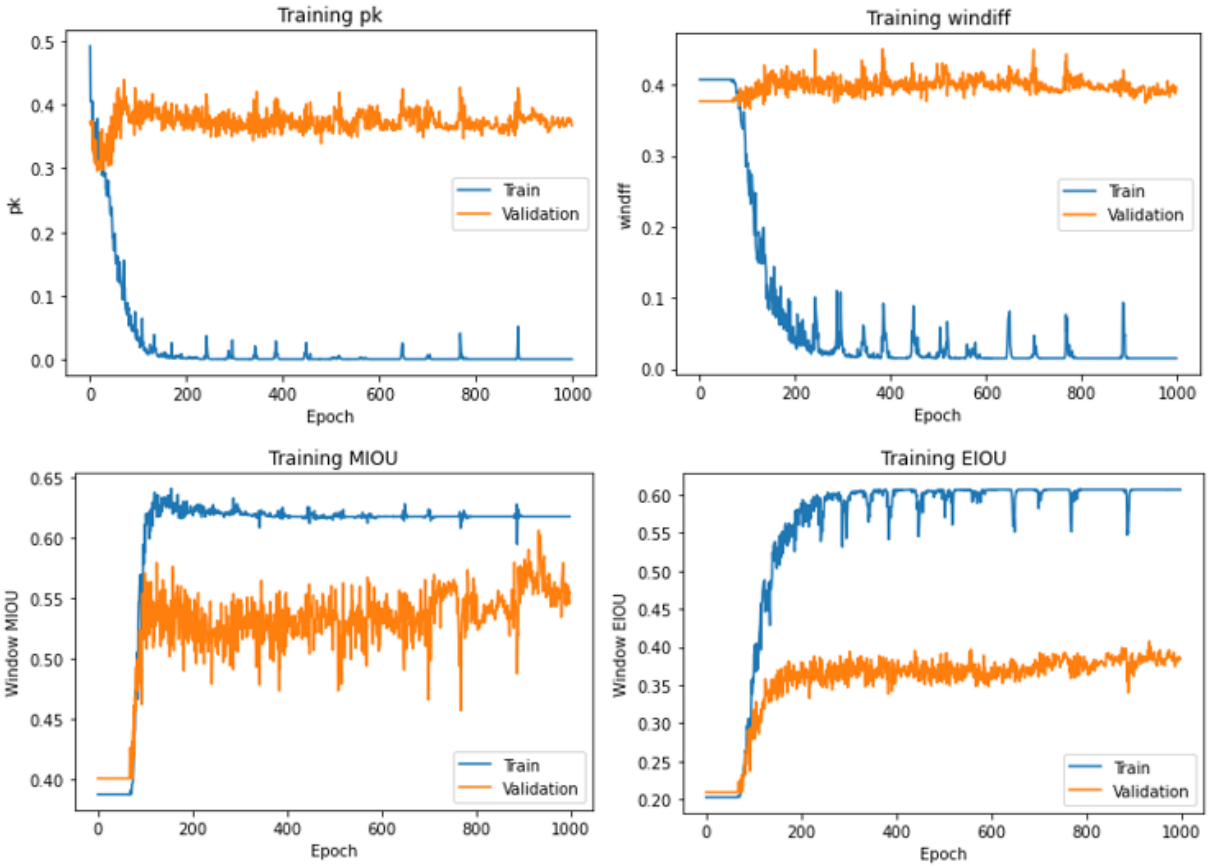


Figure 4.2: SBERT LSTM training and validation curves for each metric across epochs.

retrieved for the YouTube datasets.

Table 4.3 shows that there is a domain gap between the AMI meeting dataset and the Wiki-727k and YouTube datasets. Training on any dataset, but AMI has worse performance on AMI. Of the other datasets, training on YT-Programming performs best on AMI, which shows that a model trained on programming tutorial videos from YouTube can effectively chapter meetings better than when trained on Wikipedia articles. CS BERT and CS BERT LSTM generalizes poorly to AMI when trained on Wikipedia and YouTube. The reasoning for this could be the models being too large or the local sentence representation varying heavily between domains.

Pre-training and then fine-tuning did not have a large impact on performance. For CS

Models	Trained On	Tested On	$P_k \downarrow$	$WD \downarrow$
SBERT LSTM [37]	Wiki-727k	Wiki-727k (train)	0.3354	0.4687
SBERT LSTM [37]	Wiki-727k	Wiki-727k (val)	0.3376	0.4713
SBERT LSTM [37]	Wiki-727k	Wiki-727k (test)	0.3369	0.4700
SBERT LSTM [37]	YT-Programming	YT-Programming (train)	0.0542	0.0983
SBERT LSTM [37]	YT-Programming	YT-Programming (val)	0.3593	0.4773
SBERT LSTM [37]	YT-Programming	YT-Programming (test)	0.3476	0.5339

Table 4.2: Chaptering performance of train, val, and test sets after training on Wiki-727k and YT-Programming.

Models	Trained On	$P_k \downarrow$	$WD \downarrow$	$MIoU \uparrow$	$EIoU \uparrow$
SBERT LSTM [37]	AMI	0.3596	0.4281	0.4547	0.3909
SBERT LSTM [37]	ICSI	0.4428	0.4801	0.4454	0.3466
SBERT LSTM [37]	ICSI, AMI	**0.3549**	0.4284	0.4363	0.3906
SBERT LSTM [37]	Wiki-727k	0.4262	0.4703	0.4805	0.3581
SBERT LSTM [37]	Wiki-727k, AMI	0.3599	0.4337	0.4429	**0.4096**
SBERT LSTM [37]	YT-Programming	0.3928	0.4308	**0.5337**	0.3462
SBERT LSTM [37]	YT-Programming, AMI	0.3840	0.4568	0.4667	0.3939
SBERT LSTM [37]	YT-Design	0.3992	0.4211	0.5144	0.2971
SBERT LSTM [37]	YT-Programming-Design	0.3704	**0.3936**	0.5179	0.2686
CS BERT [23]	AMI	**0.3265**	**0.4031**	0.4601	**0.3984**
CS BERT [23]	Wiki-727k	0.4063	0.4063	0.3643	0.1883
CS BERT [23]	YT-Programming	0.4076	0.4076	0.3609	0.1889
CS BERT [23]	YT-Programming, AMI	0.3454	0.4048	**0.5286**	0.3881
CS BERT LSTM [23]	AMI	**0.3454**	0.4048	**0.5286**	**0.3881**
CS BERT LSTM [23]	Wiki-727k	0.4022	0.4079	0.3852	0.1969
CS BERT LSTM [23]	Wiki-727k, AMI	0.3847	**0.4028**	0.4559	0.2309
CS BERT LSTM [23]	YT-Programming	0.3894	0.4078	0.4789	0.2773
CS BERT LSTM [23]	YT-Programming, AMI	0.3755	0.4155	0.5003	0.3420

Table 4.3: Chaptering performance of transfer learning on AMI dataset. The dataset the models were pre-trained on is listed first followed by the dataset it was fine-tuned on.

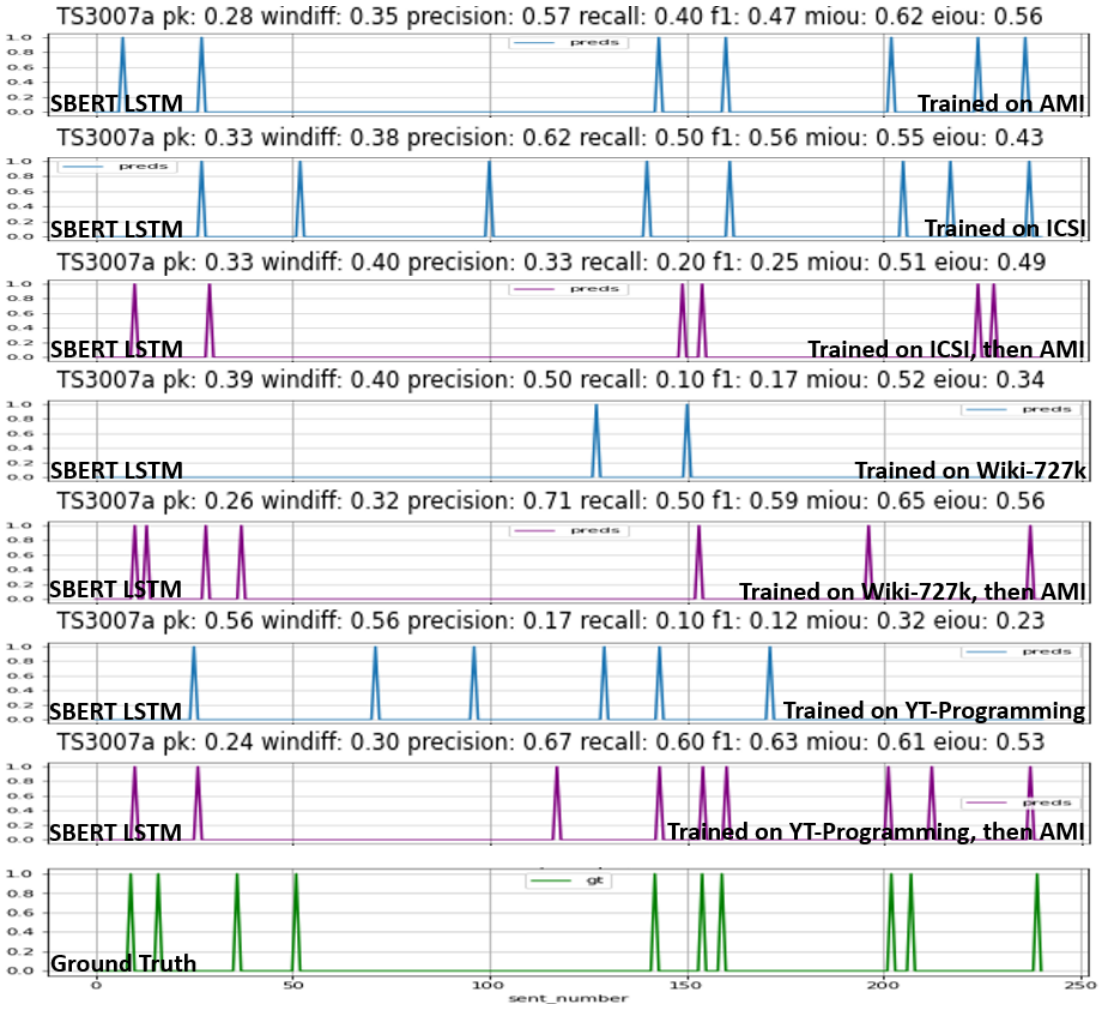


Figure 4.3: Predictions on sample meeting TS3007a after transfer learning on various datasets.

BERT and CS BERT LSTM, it hindered performance compared to just training on the target domain.

4.3 Effect of Label Smoothing and Augmentation

Label smoothing when training on AMI did not improve in P_k and WD but did improve in $EIoU$, as shown in Table 4.4. Label smoothing when training on YT-Programming helped the performance in its domain, shown in Table 4.5, but did not improve on AMI, while label smoothing was completely ineffective when training on Wiki-727k. This is likely because YouTube videos contain noisier chapter labels compared to Wikipedia articles, which would have clean chapter labels.

Using augmentation yielded a worse score in P_k and WD, but a better score in $EIoU$. Employing focal loss instead of weighted BCE loss had a negative impact on performance. As shown in the training and validation curves for $EIoU$ in Figure 4.4, none of these techniques helped reduce the amount of overfitting, it just took more epochs for the training data to overfit the training data.

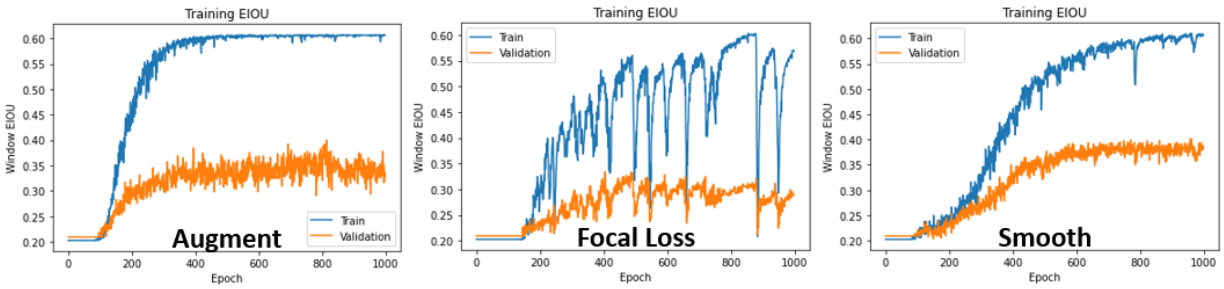


Figure 4.4: SBERT LSTM with augmentation, focal loss, and label smoothing training and validation curves for $EIoU$ across epochs.

4.4 Effect of Ensemble

Using an ensemble classification method, a better score of WD is achieved than if using the models trained on different datasets by themselves, as shown in Table 4.6. If max pooling each model's output before ensemble classification, a better score of P_k and $EIoU$ is achieved. Figure 4.6 visualizes how the precision of the chapters increased on a sample meeting as a result of the ensemble classification.

Models	Trained On	$P_k \downarrow$	$WD \downarrow$	$MIoU \uparrow$	$EIoU \uparrow$
SBERT LSTM [37]	AMI	0.3596	0.4281	0.4547	0.3909
SBERT LSTM [37]	AMI (augment)	0.3759	0.4550	0.4120	0.3938
SBERT LSTM [37]	AMI (focal loss)	0.4061	0.4610	0.4448	0.3583
SBERT LSTM [37]	AMI (smooth)	0.3864	0.4519	0.4630	0.4141
SBERT LSTM [37]	YT-Programming	0.3928	0.4308	0.5337	0.3462
SBERT LSTM [37]	YT-Programming (smooth)	0.4026	0.4232	0.4398	0.2598
SBERT LSTM [37]	Wiki-727k	0.4262	0.4703	0.4805	0.3581
SBERT LSTM [37]	Wiki-727k (smooth)	0.4325	0.4760	0.4602	0.3312

Table 4.4: Performance of augmentation, label smoothing, and focal loss on AMI dataset.

Models	Trained On	Tested On	$P_k \downarrow$	$WD \downarrow$
SBERT LSTM [37]	Wiki-727k	Wiki-727k	0.3369	0.4700
SBERT LSTM [37]	Wiki-727k (smooth)	Wiki-727k	0.4191	0.4507
SBERT LSTM [37]	YT-Programming	YT-Programming	0.3476	0.5339
SBERT LSTM [37]	YT-Programming (smooth)	YT-Programming	0.3334	0.5087

Table 4.5: Inter-domain performance after training on Wiki-727k and YT-Programming with label smoothing.

Models	Trained On	$P_k \downarrow$	$WD \downarrow$	$MIoU \uparrow$	$EIoU \uparrow$
SBERT LSTM [37]	AMI	0.3596	0.4281	0.4547	**0.3909**
SBERT LSTM [37]	Wiki-727k	0.4262	0.4703	0.4805	0.3581
SBERT LSTM [37]	YT-Programming	0.3928	0.4308	**0.5337**	0.3462
SBERT LSTM [37]	Ensemble	0.3784	**0.3911**	0.4572	0.2368
SBERT LSTM [37]	Ensemble w/ MaxPool	**0.3581**	0.4063	0.5152	0.3725

Table 4.6: Performance of ensemble classification on AMI dataset.

4.5 Run Time

The run time of each model is shown in Table 4.1. CS BERT is the fastest method, although if SBERT embeddings are cached ahead of time, then SBERT LSTM is able to process the entire AMI training dataset in less than a second. The bottleneck for all models

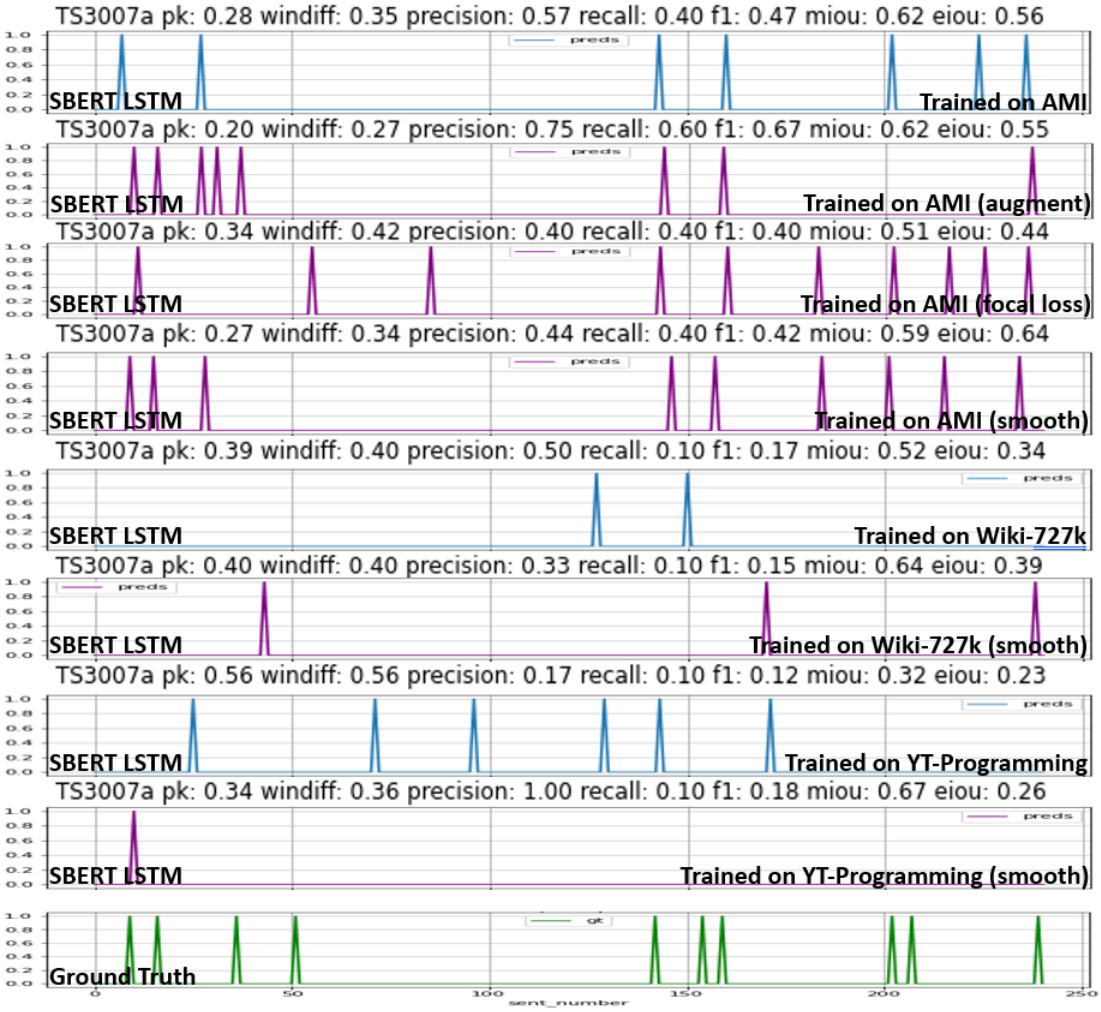


Figure 4.5: Predictions on sample meeting TS3007a after employing label smoothing, augmentation and focal loss.

is computing the BERT embeddings.

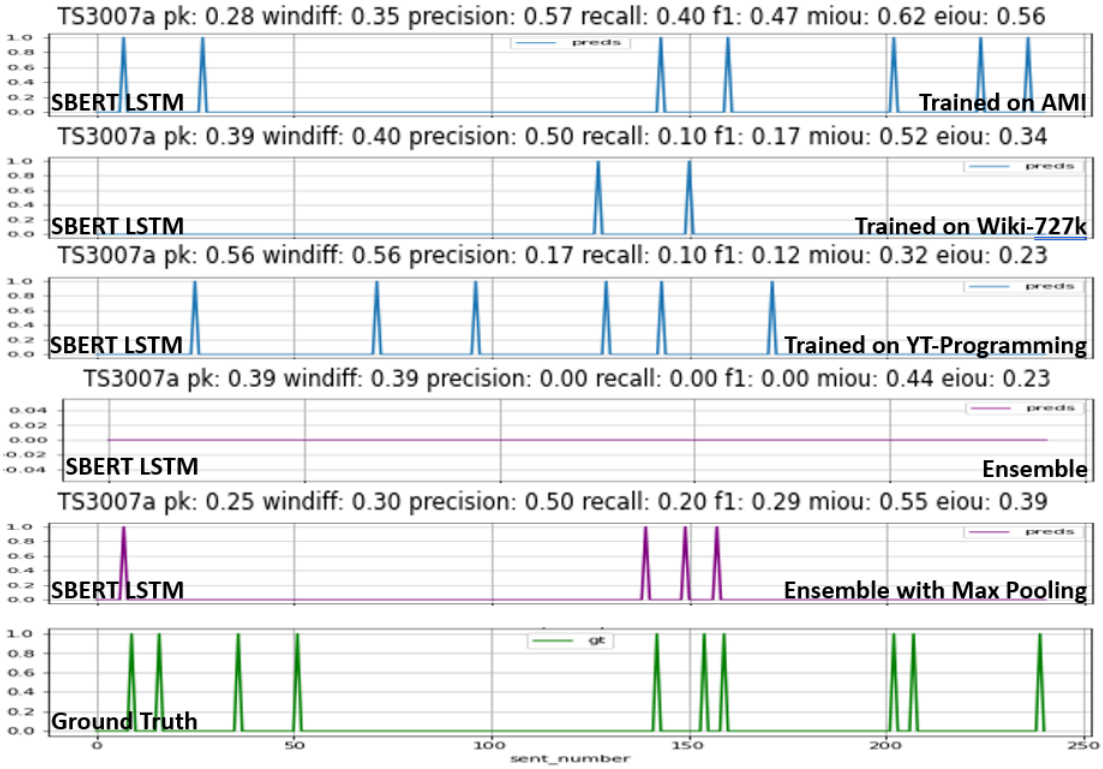


Figure 4.6: Chaptering predictions with ensemble classification on sample meeting TS3007a.

Models	Training (s)	Testing (s)
SBERT Text-Tiling [32]	N/A	47.70
SBERT LSTM [37]	39.69	42.34
SBERT LSTM [37] (cached SBERT embeddings)	0.27	0.14
CS BERT [23]	19.05	16.08
CS BERT LSTM [23]	19.41	16.63

Table 4.7: Run time of different chaptering methods on AMI training dataset.

4.6 Critique of Metrics and Suggestion for an Improved Metric

Upon investigation, P_k and WD may not be reliable metrics. The windowing function does not actually have the effect of relaxing near misses as it was originally intended [27]. As shown in Figure 4.7 and Figure 2.1, any window size of k produces the same amount of error regardless of the near miss. Also, both metrics considering error at every sentence index makes them not so different from a standard accuracy metric, which is not ideal in a task where there are far more negative labels than positive labels.

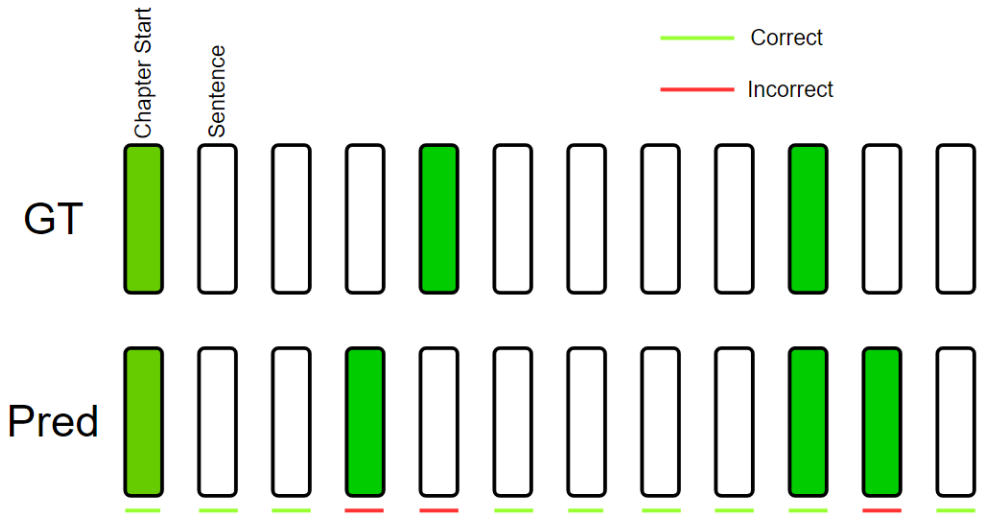


Figure 4.7: A visual illustration of the sliding window evaluation for the P_k and WD metric when window size k is 1.

A potential improvement would be to only consider the window around the chapter starts in the ground truth and prediction as shown in Figure 4.8. This 'revised' P_k (rP_k) can be represented with the following equation:

$$rP_k = \frac{1}{b(y_1, b_N) + b(p_1, p_N)} \sum_{i=1}^{N} ((y_i|p_i)(b(y_{i-k/2}, y_{i+k/2}) > 0) \neq (b(p_{i-k/2}, p_{i+k/2}) > 0)) \tag{4.1}$$

while the 'revised' WD (rWD) can be represented with the equation:

$$rWD = \frac{1}{b(y_1, y_N) + b(p_1, p_N)} \sum_{i=1}^{N} ((y_i|p_i)|b(y_{i-k/2}, y_{i+k/2}) - b(p_{i-k/2}, p_{i+k/2})| > 0) \quad (4.2)$$

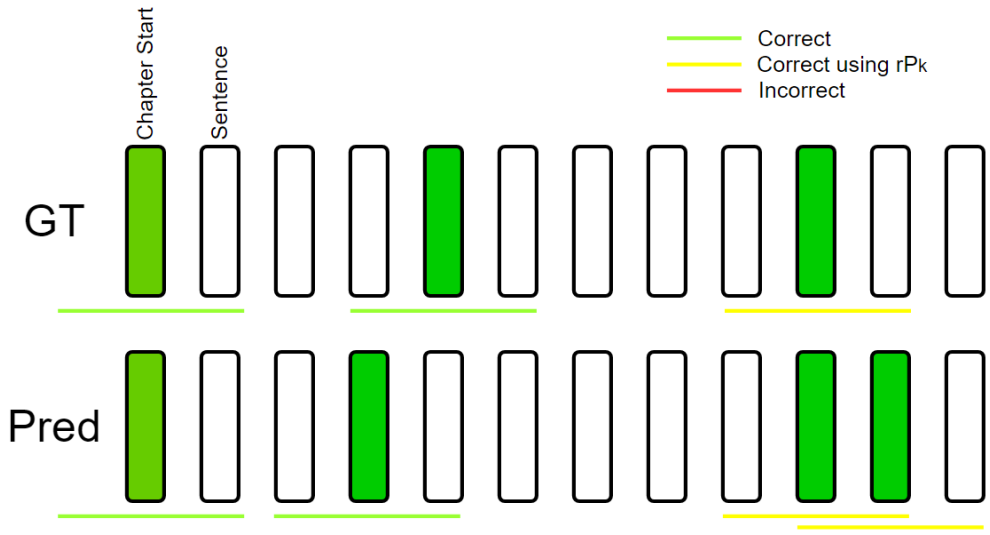


Figure 4.8: A visual illustration of the revised window evaluation. Only the area around sentences labelled as chapters in the ground-truth and predictions counts for evaluation, instead of at every sentence index, as displayed in Figure 2.1.

These revised metrics more closely resemble a precision and recall metric rather than accuracy. Results with these revised metrics can be found in Table 4.8.

Many of the experiments can be re-contextualized under these revised metrics, although it is worth re-iterating that the models were picked to be the best-performing models for *WD* and *EIoU* on the validation set and not these new metrics. With these metrics, SBERT LSTM has the best scores of all the chaptering methods tested and pre-training on a larger corpus like Wikipedia before fine-tuning is shown to be somewhat effective with this metric, along with label smoothing of the AMI dataset. Ensemble classification performs very poorly under this metric, likely due to recall being higher as it is more selective with the chapter predictions.

Models	Trained On	$rP_k \downarrow$	$rWD \downarrow$
None	N/A	1.000	1.000
Even	N/A	0.5203	0.6870
Random	N/A	0.6301	0.7600
ChatGPT 3.5	N/A	0.4645	0.6377
SBERT Text-Tiling [32]	N/A	0.8365	0.8734
SBERT LSTM [37]	AMI	**0.4437**	**0.6290**
CS BERT [23]	AMI	0.4502	0.7012
CS BERT LSTM [23]	AMI	0.5130	0.6988
SBERT LSTM [37]	AMI (augment)	**0.4283**	0.6439
SBERT LSTM [37]	AMI (smooth)	**0.4189**	**0.6183**
SBERT LSTM [37]	ICSI	0.5940	0.7142
SBERT LSTM [37]	ICSI, AMI	**0.4277**	0.6562
SBERT LSTM [37]	Wiki-727k	0.6101	0.7458
SBERT LSTM [37]	Wiki-727k (smooth)	0.7822	0.8336
SBERT LSTM [37]	Wiki-727k, AMI	**0.4060**	**0.6237**
SBERT LSTM [37]	YT-Programming	0.6549	0.8277
SBERT LSTM [37]	YT-Programming (smooth)	0.7822	0.8336
SBERT LSTM [37]	YT-Programming, AMI	**0.4368**	0.6485
SBERT LSTM [37]	YT-Design	0.8202	0.9324
SBERT LSTM [37]	YT-Programming-Design	0.7893	0.9146
SBERT LSTM [37]	Ensemble w/ Max Pool	0.5263	0.7436

Table 4.8: Performance of different chaptering methods on AMI dataset with revised metrics.

Chapter 5

Conclusions and Future Work

Current state-of-the-art supervised learning techniques for topic segmentation (Section 3.2) perform well for chaptering meetings (Section 4.1). The favourable technique largely depends on the metric used, although SBERT LSTM has benefits in that it is smaller (Section 3.2), much easier to train (Section 4.5) and is more suitable for domain transfer (Section 4.2). Training BERT layers did not prove to be very helpful, pre-trained SBERT is already an adequate sentence representation for the task (Section 4.1).

Domain transfer is a challenge for topic segmentation, as a model has to be trained in the same domain to have good results. However, more niche domains will have less training data making it susceptible to overfitting, which couldn't be solved by the data augmentation techniques employed in this thesis (Section 4.3). It was shown that it is not beneficial to pre-train on a large corpus of a different domain before fine-tuning on the target meeting domain (Section 4.2). Fortunately, only a few videos in the target domain is sufficient to train a model to still chapter effectively, despite the large amount of overfitting. The custom YouTube dataset curated did not prove to be more effective for pre-training than the established Wiki-727k, despite being closer in domain (Section 4.2). However, this could be a result of there not being enough videos collected as training had overfit on the YouTube dataset. More videos should be obtained in the future to fully determine whether training on YouTube videos for meeting chaptering is effective. Furthermore, the effect of pre-training on the synthetic datasets discussed in Section 2.1 should also be explored for meeting chaptering in future research.

Despite pre-training on a large corpus not being beneficial, having different models trained on different datasets and used for ensemble classification was shown to have promising results (Section 4.4). This approach mimics how human annotators for some datasets

have to have some agreement on chapters due to the subjectivity of the task, and exploring this idea further could be a promising area of further exploration.

Given the subjective nature of the problem, there can never be a perfect prediction model, however, it is important that metrics properly reflect user expectations in order to be able to be able to evaluate improvements to chaptering techniques. The current common metrics used for topic segmentation, P_k and WD, do not properly account for the 'near miss' (Section 4.6), despite being designed to, and future work in topic segmentation should consider retiring these metrics as the primary metrics for topic segmentation in favour of new metrics that more closely match user expectations.

References

[1] Pinkesh Badjatiya, Litton J. Kurisinkel, Manish Gupta, and Vasudeva Varma. *Attention-based neural text segmentation*, volume 10772 LNCS, pages 180–193. 2018.

[2] Joe Barrow, Rajiv Jain, Vlad Morariu, Varun Manjunatha, Douglas Oard, and Philip Resnik. A joint model for document segmentation and segment labeling. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL 2020, pages 313–322, Online, July 2020. Association for Computational Linguistics.

[3] Doug Beeferman, Adam Berger, and John Lafferty. Statistical Models for Text Segmentation. *Machine Learning*, 34:177–210, 1999.

[4] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2018, pages 5016–5026, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[5] Jean Carletta. Training, test and evaluation sets for the ami corpus, 2006. Available at https://groups.inf.ed.ac.uk/ami/corpus/datasets.shtml.

[6] Google Search Central. Video (videoobject, clip, broadcastevent) structured data, 2024. Available at https://developers.google.com/search/docs/appearance/structured-data/video.

[7] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, volume 1 of *NAACL-HLT 2019*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[8] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS 2000, page 1–15, Cagliari, Italy, June 2000. Springer-Verlag.

[9] Jacob Eisenstein and Regina Barzilay. Bayesian unsupervised topic segmentation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2008, pages 334–343, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[10] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL 2017, pages 37–49, Saarbrücken, Germany, August 2017.

[11] Reshmi Ghosh, Harjeet Singh Kajal, Sharanya Kamath, Dhuri Shrivastava, Samyadeep Basu, and Soundararajan Srinivasan. Topic Segmentation in the Wild: Towards Segmentation of Semi-structured & Unstructured Chats. In *Proceedings of the 2nd workshop on Efficient Natural Language and Speech Processing*, ENLSP-II, pages 1–9, New Orleans, Louisiana, December 2022. NeurIPS.

[12] Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. Unsupervised text segmentation using semantic relatedness graphs. In Claire Gardent, Raffaella Bernardi, and Ivan Titov, editors, *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130, Berlin, Germany, August 2016. Association for Computational Linguistics.

[13] Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. Topical-chat: Towards knowledge-grounded open-domain conversations. In *Proceedings of the 20th Annual Conference of the International Speech Communication Association*, INTERSPEECH 2019, pages 1891–1895, Graz, Austrira, September 2019.

[14] Alexander Gruenstein, John Niekrasz, and Matthew Purver. *Meeting Structure Annotation*, volume 39, page 247–274. Springer, 2008.

[15] Marti Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[16] YouTube Help. Video chapters, 2020. Available at https://support.google.com/youtube/answer/9884579?hl=en.

[17] Nicole Herskowitz. Microsoft teams premium: Cut costs and add ai-powered productivity, 2024. Available at https://www.microsoft.com/en-us/microsoft-365/blog/2023/02/01/microsoft-teams-premium-cut-costs-and-add-ai-powered-productivity/.

[18] Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. A Review of Deep Transfer Learning and Recent Advancements. *Technologies*, 11(2), 2023.

[19] Adam L. Janin, Don Baron, Jane Edwards, Daniel P. W. Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. The icsi meeting corpus. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1 of *ICASSP 2003*, pages I–I, Hong Kong, China, April 2003.

[20] Lei Ji, Chenfei Wu, Daisy Zhou, Kun Yan, Edward Cui, Xilin Chen, and Nan Duan. Learning Temporal Video Procedure Segmentation from an Automatically Collected Large Dataset. In *Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision*, WACV 2022, pages 2733–2742, Waikoloa, Hawaii, January 2022.

[21] Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text segmentation as a supervised learning task. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, NAACL-HLT 2018, pages 469–473, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[22] Tsung Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2018.

[23] Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. Text segmentation by cross segment attention. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2020, pages 4707–4716, Online, November 2020. Association for Computational Linguistics.

[24] Tengchao Lv, Lei Cui, Momcilo Vasilijevic, and Furu Wei. VT-SSum: A Benchmark Dataset for Video Transcript Segmentation and Summarization. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, volume 1 of *CIKM 2021*, pages 1–5, Queensland, Australia, November 2021. Association for Computing Machinery.

[25] Amit Maraj, Miguel Vargas Martin, and Masoud Makrehchi. *A More Effective Sentence-Wise Text Segmentation Approach Using BERT*, volume 12824 LNCS, pages 236–250. Springer International Publishing, 2021.

[26] Iain McCowan, Jean Carletta, W Kraaij, S Ashby, S Bourban, M Flynn, M Guillemot, T Hain, J Kadlec, V Karaiskos, and Others. The AMI Meeting Corpus. In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, Wageningen, Netherlands, August 2005.

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

[28] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A local-to-global approach to multi-modal movie scene segmentation. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR 2020, pages 10143–10152, Seattle, Washington, June 2020.

[29] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2019, Hong Kong, China, November 2019. Association for Computational Linguistics.

[30] Ken Sallee. Does this actually work? — youtube auto chapters experiment, 2021. Available at https://www.youtube.com/watch?v=ekYzHqrpd48.

[31] Sarah. [previous experiments] youtube test features and experiments, 2022. Available at https://support.google.com/youtube/thread/162258829/previous-experiments-youtube-test-features-and-experiments?hl=en.

[32] Alessandro Solbiati, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. Unsupervised Topic Segmentation of Meetings with BERT Embeddings. 2021.

[33] Zhiyi Song, Stephanie Strassel, Haejoong Lee, Kevin Walker, Jonathan Wright, Jennifer Garland, Dana Fore, Brian Gainor, Preston Cabe, Thomas Thomas, Brendan

Callahan, and Ann Sawyer. Collecting natural SMS and chat conversations in multiple languages: The BOLT phase 2 corpus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC 2014, pages 1699–1704, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[34] Tiago Henrique Trojahn and Rudinei Goularte. Temporal video scene segmentation using deep-learning. *Multimedia Tools and Applications*, 80:17487–17513, 2021.

[35] Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing Long Han, and Yang Tang. A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136, 2023.

[36] Linzi Xing and Giuseppe Carenini. Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring. In Haizhou Li, Gina-Anne Levow, Zhou Yu, Chitralekha Gupta, Berrak Sisman, Siqi Cai, David Vandyke, Nina Dethlefs, Yan Wu, and Junyi Jessy Li, editors, *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL 2021, pages 167–177, Singapore and Online, July 2021. Association for Computational Linguistics.

[37] Linzi Xing, Brad Hackinen, Giuseppe Carenini, and Francesco Trebbi. Improving context modeling in neural topic segmentation. In Kam-Fai Wong, Kevin Knight, and Hua Wu, editors, *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, AACL 2020, pages 626–636, Suzhou, China, December 2020. Association for Computational Linguistics.

[38] Weiqun Xu, Jean Carletta, Jonathan Kilgour, and Vasilis Karaiskos. Coding Instructions for Topic Segmentation of the AMI Meeting Corpus, 2005. Available at https://groups.inf.ed.ac.uk/ami/corpus/Guidelines/TopicSegmentationGuidelines.pdf.

[39] Yi Xu, Hai Zhao, and Zhuosheng Zhang. Topic-Aware Multi-turn Dialogue Modeling. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, volume 16 of *AAAI 2021*, pages 14176–14184, Online, 2021.

[40] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir

Radev. QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT 2021, pages 5905–5921, Online, June 2021. Association for Computational Linguistics.

APPENDICES

Appendix A

Chapters in a Sample Transcript

In this appendix, the chapters predicted by different methods as well as ground truth will be inserted in the transcript of sample AMI meeting TS3007a [26]:

```
My name is Danny Wolfs.
 this is the agenda for today.
 first a little opening.
 I will introduce myself, and I think it's very good to introduce yourself.
 then  a little bit of acquaintance, acquaintance to to to ourselves.
So  we get to know each other.
 that will be done with a tool training from the these two smart boards.
============SBERT LSTM Chapter============
Then the project plan.
What we're going to do, and how we're going to do it.
============True Chapter============
 and discussion about that and a little closing at the end.
============CS BERT LSTM Chapter============
Okay , my name is Danny Wolfs.
I'm the Project Manager.
I'm the User Interface Designer.
User interface, okay.
I'm the Industrial Designer.
============CS BERT Chapter============
 my function is the Marketing Expert.
============True Chapter============
First a little about the project aim.
============ChatGPT Chapter============
```

the the the aim is to make a new remote control.
==========CS BERT Chapter==========
maybe you have read read the website.
It's a very , yeah, very ambitious company.
They they wanna do something else.
I there must be a new remote control.
==========CS BERT Chapter==========
first of all it must be original, and trendy.
That's two things really close to each other.
But at the same time user-friendly.
And they have Yeah, that's very important for them.
there are three stages.
==========SBERT LSTM==========
There is a functional design.
==========CS BERT Chapter==========
So what are we going to to do?
What are we going to make kind of functions in the remote?
And why are we going to do it?
Then the conceptual design.
How are going to do it?
And that's really global.
because at the detailed design, how, part two, we go to dig in really about how the the the technical of If it's it's possible technical-wise.
==========CS BERT Chapter==========
each stage is is broken up in two two stages, individual work and a meeting.
==========True Chapter==========
So it's it's very straightforward.
==========ChatGPT Chapter==========
Okay, the tool training.
We have two smart boards.
This one is for the presentations, the PowerPoint presentations or the Word presentation of whatever you you had.
and this is only for drawing.
So we must let it stand on this this programme.
This is called a smart board thing Yeah, it speaks for itself.
and as you may have heard, the docents in the shared folder can be showed on this screen.
Not in the the My Docents.
So if you wanna show something, put it in the shared folder.

This is very straightforward, with the save, the print, the undo, the blank, the select, the pen.
Well, I don't gonna explain it all, because I think how it works.
we must not forget everything we draw on here, all must be saved.
We we may not delete anything.
So if you have drawn something, save it.
==========True Chapter==========
That's a very important thing.
little little kinda exercise to know each other.
At the white board on the left.
Every every one of us must draw our favourite animal, and tell tell us why we had chosen that animal.
important is that we use different colours, and different pen widths.
this exercise, did the company board tell you to do it, or did you just make it up yourself?
It's it's I I I must do it.
It's part of the introduction, okay.
'Cause we really don't know each other, and it's kinda new.
So getting used to each other, we can have a little fun then, before we dig in really to the hard stuff.
Would you start with drawing your favourite animal?
I don't know really how it works.
But maybe you can show us first?
==========CS BERT Chapter==========
Drawing goes with this thing.
Do not touch your hand on this little thingy here.
You you get electrocuted or It's Pictionary?
So, You must push a little Good.
Because And wait wait a few seconds.
It's not fully real-time, so watch it.
Well I'm gonna paint in the red.
That's the background colour.
Yeah, that's the one.
My favourite animal h?
Yeah, you can guess what it is.
The the one who says it first gets a raise.
You don't see it at the ears?
Mm yeah, I have it at home.
You have an orc at home?
'Cause we have a cat at home and he's called Simba.

’Cause he looks the the the lion from The Lion King.
So we found it kinda cool to name it after a lion.
He’s happy with us, so he’s smiling.
He does have body H.
Because we have we have twenty five minutes.
Remember you use different colours, and different pen widths.
Okay, who wants to go next?
So choose a colour, choose a pen width and draw a Yeah.
You don’t have to change the colour and the pen width during the drawing.
Or You have to draw push hard on the pen or .
By the way, why was your cat red?
Because my cat is red at home.
And I have red hair, so must be red.
It’s a very bloody cat.
’Cause I d Teenage Mutant Hero Turtles.
You watched it a lot?
You watched it a lot?
It’s inside its shell.
You’ll be finished sooner.
It’s a scared turtle.
Okay, Very artistic.
Jurgen, you want to go next?
And then you have the the current colour, you can change.
Tim had Tim, what kinda line width did you have?
the big lines were nine.
I just thought I’d pick the easiest one.
You have a dog at home?
Well, we had a dog, a few years ago.
, it’s It’s pretty good An artist.
And and it, yeah, when it died we didn’t get a new one or something.
But a Graphical User Designer, so Hey.
You have an artistic inner middle.
Think you picked the wrong function.
So I think you can see it’s really a easy programme to use.
Not difficult at all.
That’s enough, thanks.
After a cat, a turtle and a dog.
I think he’s gonna draw an elephant.
I figure I should do something that, but I’m gonna do something much more difficult.

, he is the artistic design.
I'm gonna design a remote remote control animal.
Remote control animal.
Well with the interface, it might be easier to to draw here and display there .
That that might be easier.
But at the other hand, a pen that is easy to hold in your hand, and I think it's easier to draw.
Better to draw with a with But what is he ?
With a pen than with a mouse mouse.
Than on the, with Yeah, I I mean on here, drawing drawing .
And then displaying on screen, but Too expensive, yeah.
with this paper it's too too expensive.
Do you have a rabbit at home?
It's a rabbit with broken legs?
Is it a white rabbit It's the white rabbit from The Matrix.
There, the white green rabbit.
He's a little bit stoned there.
I figured this is a pretty good impression of a rabbit.
Finishing touch and then we're going further.
Have you been counting the time?
Where does the pen go?
Just I figured the rabbit was actually the most impressive.
Well, I think the dog is the the most artistic.
Don't choose for youself.
==========True Chapter==========
Okay, now we're gonna dig into the to the serious stuff.
==========SBERT LSTM Chapter==========
It's pretty abstract.
==========CS BERT Chapter==========
the selling price for the remote will be twenty five Euro, and the production cost may not be more than twenty and a half Euro.
So from my point of view, I don't think it's gonna be very very high tech, high definition, ultra modern kinda remote, for twelve twelve and a half Euro.
==========CS BERT Chapter==========
the profit we must make with the new remote is fifty million Euro.
We have to sell a lot of I think when the selling price is twenty five, you got two million, two million remotes.
Yeah, how much is it?
Hundred million remotes or something?
Two million, yeah, two million.

But our marketing range is , market range is international.
So we have virtually the whole world we can sell we can sell our remotes to.
At least that countries which have a television.
==========True Chapter==========
==========ChatGPT Chapter==========
So now it's time for us to going to discuss a little things.
You can think about experience with a remote control yourself, at home.
What you think might be a useful new feature.
What what can distinguish our new trendy remote control from all the others.
==========CS BERT Chapter==========
so let's let's discuss a little.
==========True Chapter==========
I'm gonna join you at the table.
==========SBERT LSTM Chapter==========
==========CS BERT LSTM Chapter==========
Well what what's the most important thing at a remote control?
well I think the most important thing of a remote control is that you can switch channels.
==========CS BERT Chapter==========
And my opinion is you should keep it as basic as possible.
==========ChatGPT Chapter==========
Is it possible to open pen drawings in this on this screen?
==========CS BERT Chapter==========
==========CS BERT LSTM Chapter==========
So not a not a remote control who which can can be used for television and a D_V_D_ and radio and Or just only If you have them on I can Okay.
but I have some points.
==========CS BERT LSTM Chapter==========
Can I show them on the on the big screen?
, in case you want it This is a dead kind of fly.
Between the the the, yeah, the the Yeah, The screens.
Only All the drawings go there, at the left , that pen drawings.
but which The ones we made on the , only in Word, okay.
no, I think when it is in Word and you have saved it in the Shared Docents folder, you can show it there.
Okay, I have some points from marketing point of view.
just the standard thing things intuitive, small, fairly cheap.
it's pretty cheap, twenty five Euros.
I think, it doesn't have to matter which brand your T_V_ or other thing is.
Okay, I'll wrap it up quickly.

I personally think it has to be multi-purpose.
==========CS BERT LSTM Chapter==========
most of the remote remote controls are just for one purpose.
And by making it multi-purpose, it has a new feature, adds a new feature to the market, and distinguish from from current products.
maybe some other technology than infrared.
I rather find it very annoying , when someone is standing in front of the T_V_ then you can't switch it.
think about sending it over radio waves or bluetooth.
That might be a little bit expensive.
And something an L_C_D_ screen.
I said here Maybe it's easy.
It's nice as an added feature feature, that , when you're on a certain channel, you can see on the L_C_D_ screen what programmes are coming up or From my point of view, yeah.
So it be a multi-purpose very technically high remote?
Yeah, it must be really innovative, technical-wise?
Yeah, it has to be Yeah, our company is very good in making new innovative things.
So yeah, I I agree with you.
So To the current market.
We And and the price.
So we must focus on things who are really really add something to to Yeah.
Look, you got some cheap remote controls there.
They just Yeah, you got a dozen of 'em.
But when you enter a new market with a remote control and wanna gain market share you have to do something special, I think.
But we have to keep an eye that it's at the beginning of such a project, it's it 's it's very cool to talk about, well, this would be cool, that would be cool.
but we must not lose sight of the the user friendly Yeah okay.
But it's But but this is just from marketing aspect.
I don't know anything about user interface or design.
And that's because we have him.
==========True Chapter==========
==========SBERT LSTM Chapter==========
Okay, next meeting will start in thirty minutes.
So you will have individual actions where I prese will be some feedback, via the the mail.
the the the Industrial Designer has to look at the working design.

==========CS BERT LSTM Chapter==========
the User Interface Designer has to look at the technical functions.
So that's the thing we discussed.
==========True Chapter==========
one thing , we must first agree on what we're going to going to make.
Do we Are we going to use it it for multiple systems?
==========CS BERT LSTM Chapter==========
Or We should have some agreement on that before we Okay.
==========CS BERT Chapter==========
Mm I I don't think we have to be, we have to agree on that.
I think that's a Yeah.
I figure we could get back to it on the next meeting actually.
That's a phase further.
just make some mock-ups, some some general ideas.
==========CS BERT Chapter==========
And and then we can plan Yeah.
We can plan further, I think.
But maybe, because you are working on the user requirements, you are working on the technical functions, we must have a little or kinda How do you call it?
Consensus on the, what we're gonna do.
a little plan on on what we're going to do.
==========ChatGPT Chapter==========
So you don't come up with the user requirements who don't fit the the the the technical functions at all.
Some basic things we we want to going to do.
I think that's well Yeah.
You're the Project Manager.
==========SBERT LSTM Chapter==========
Well yeah, if the technical functions have to be designed, I I've gotta know for what kind of machines they will be.
Or do we use it a text screen?
Or will it be with with bluetooth or Okay.
Well, that's that's really a step further.
But if you say is it one way or multi-purpose, that's a Why?
that's a same step further.
Yeah, actually it is.
Then looking at individual components, so that's actually a step further.
we all have a list of things that has to that have to be in it, or how it has to be .
And then in the next meeting we decide what it's gonna be.

Yeah, we can take it from there.
Yeah, I agree , we can take it from there.
==========SBERT LSTM Chapter==========
And then you then you can delete the the obsolete details.
So each individually individually must think on what's at his point of view is the most important.
And then we're going to fit all the pieces together the next meeting.
==========True Chapter==========
I must finish off now, so it's over.
==========ChatGPT Chapter==========
You will receive specific specific instructions by your personal coach.