Reply to Examiner No. 3

Name of Student: Andrew Koh Jin Jie

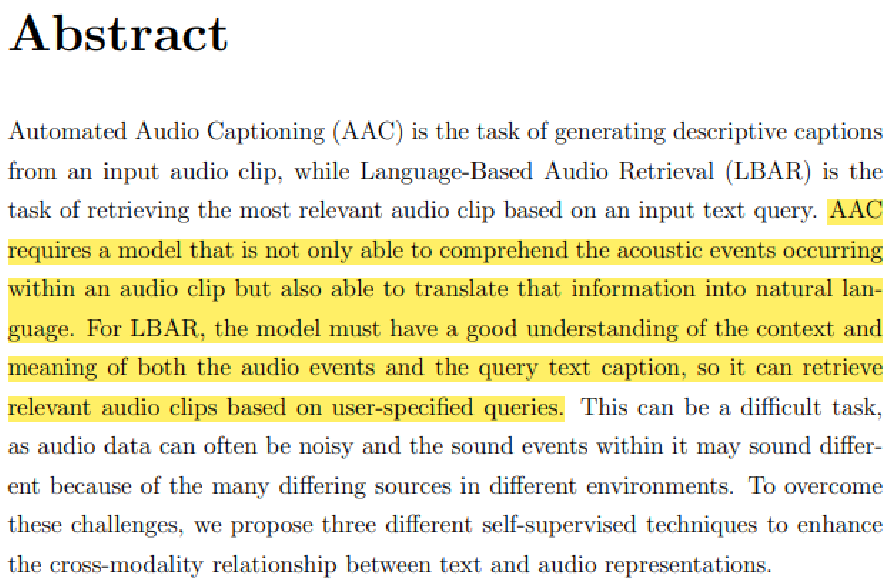
Degree: Doctor of Philosophy

Thesis Title: Audio Captioning and Retrieval with improved Cross-Modal Objectives

Examiner 3 comments

1. **Examiner's comment:** Page xiii (Abstract): When referring to automatic captioning and text-based retrieval, the thesis describes that "Both tasks require a model that is not only able to comprehend the acoustic events occurring within an audio clip, but also able to translate that information into natural language". However, it is not clear why text-based audio retrieval would require translating information into natural language. My understanding is that retrieval could be done without that, by mapping audio and text into the same domain where the relevance score is calculated. However, if retrieval requires translation information in audio to natural language, it should be clarified where exactly such information is needed.

   Response: Thank you for this comment. We have clarified the definition to clearly delineate the requirements of AAC and LBAR in the abstract.

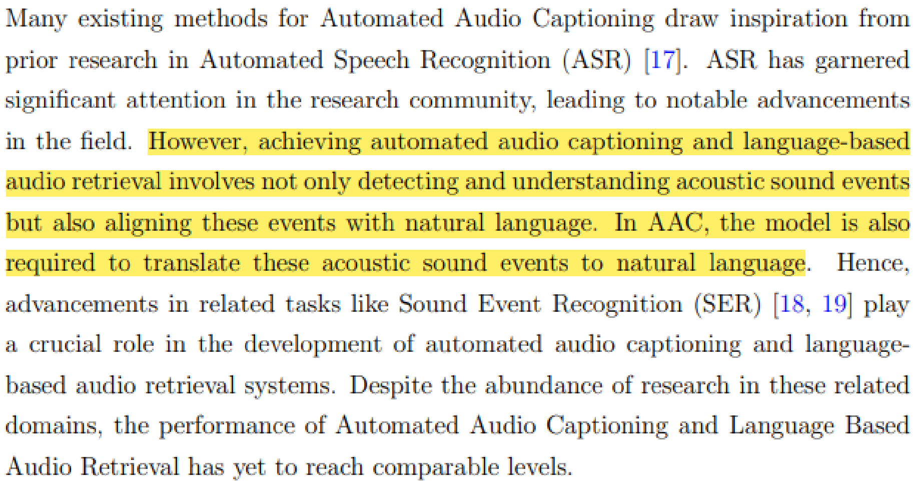


Abstract

Automated Audio Captioning (AAC) is the task of generating descriptive captions from an input audio clip, while Language-Based Audio Retrieval (LBAR) is the task of retrieving the most relevant audio clip based on an input text query. AAC requires a model that is not only able to comprehend the acoustic events occurring within an audio clip but also able to translate that information into natural language. For LBAR, the model must have a good understanding of the context and meaning of both the audio events and the query text caption, so it can retrieve relevant audio clips based on user-specified queries. This can be a difficult task, as audio data can often be noisy and the sound events within it may sound different because of the many differing sources in different environments. To overcome these challenges, we propose three different self-supervised techniques to enhance the cross-modality relationship between text and audio representations.

2. **Examiner's comment:** Page 2: The thesis describes "language-based audio retrieval involves not only detecting and comprehending acoustic sound events but also translating and aligning these events with natural language". Similarly to the above issue, it is not clear where translation to natural language is needed.

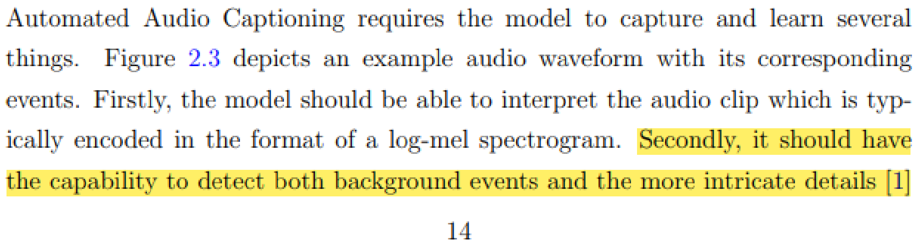
   Response: Thank you for this comment. We have clarified that both tasks require alignment between sound events and natural language. However, only audio captioning requires translation into natural language.



Many existing methods for Automated Audio Captioning draw inspiration from prior research in Automated Speech Recognition (ASR) [17]. ASR has garnered significant attention in the research community, leading to notable advancements in the field. However, achieving automated audio captioning and language-based audio retrieval involves not only detecting and understanding acoustic sound events but also aligning these events with natural language. In AAC, the model is also required to translate these acoustic sound events to natural language. Hence, advancements in related tasks like Sound Event Recognition (SER) [18, 19] play a crucial role in the development of automated audio captioning and language-based audio retrieval systems. Despite the abundance of research in these related domains, the performance of Automated Audio Captioning and Language Based Audio Retrieval has yet to reach comparable levels.

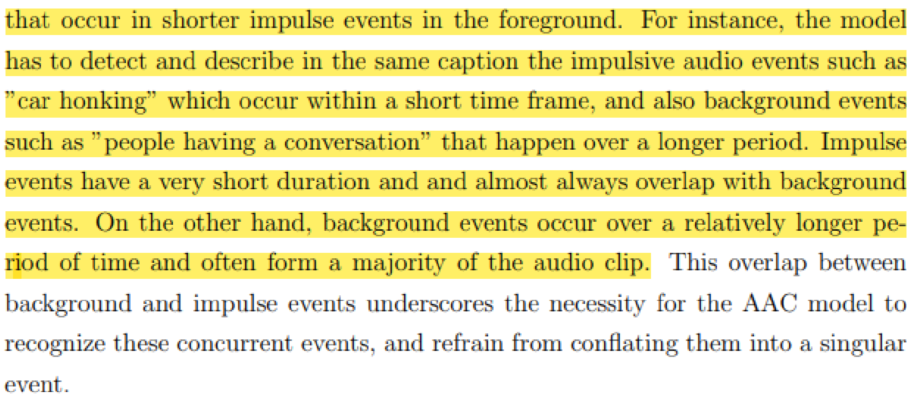
3. **Examiner's comment:** Page 15: The thesis refers to "low level events" and "high level events" but it is not clear what makes an event low or high level, and what is the difference between these two categories.

   Response: Thank you for this comment. We have replaced all mentions of low/high level sound events with impulse/background events. Impulse and background events are also described with examples given.



Automated Audio Captioning requires the model to capture and learn several things. Figure 2.3 depicts an example audio waveform with its corresponding events. Firstly, the model should be able to interpret the audio clip which is typically encoded in the format of a log-mel spectrogram. Secondly, it should have the capability to detect both background events and the more intricate details [1]

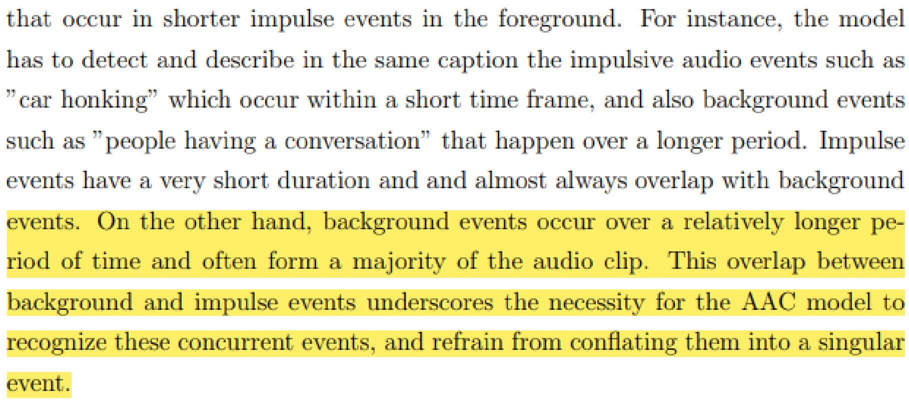
14



that occur in shorter impulse events in the foreground. For instance, the model has to detect and describe in the same caption the impulsive audio events such as ”car honking” which occur within a short time frame, and also background events such as ”people having a conversation” that happen over a longer period. Impulse events have a very short duration and and almost always overlap with background events. On the other hand, background events occur over a relatively longer period of time and often form a majority of the audio clip. This overlap between background and impulse events underscores the necessity for the AAC model to recognize these concurrent events, and refrain from conflating them into a singular event.

4. **Examiner’s comment:** Page 15: The thesis writes that “Impulse events are often short lived and and almost always overlap with background events”. It is not fully clear what ”short lived” means. Does it refer to short duration? Mentioning that impulse events overlap with background events seems weird, since I do not see how the impulsiveness of an event would be related to the background.

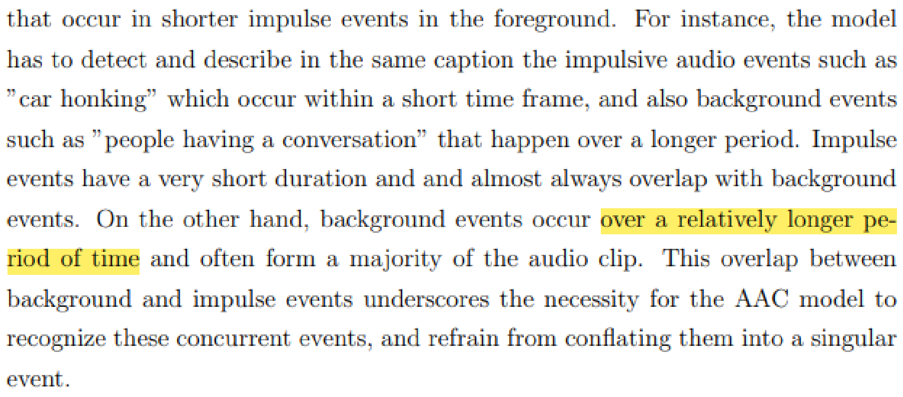
   Response: Thank you for this comment. The intention of this statement was to demonstrate that the overlap of both impulse and background sound event introduces an additional challenge to recognize the presence of two events. We have also replaced “low” and “high” with “impulse” and “background”.



that occur in shorter impulse events in the foreground. For instance, the model has to detect and describe in the same caption the impulsive audio events such as ”car honking” which occur within a short time frame, and also background events such as ”people having a conversation” that happen over a longer period. Impulse events have a very short duration and and almost always overlap with background events. On the other hand, background events occur over a relatively longer period of time and often form a majority of the audio clip. This overlap between background and impulse events underscores the necessity for the AAC model to recognize these concurrent events, and refrain from conflating them into a singular event.

5. **Examiner's comment:** Page 15: The thesis refers to "a high time frame". I assume this refers to the number of time frames, but the expression that is used is not clear.

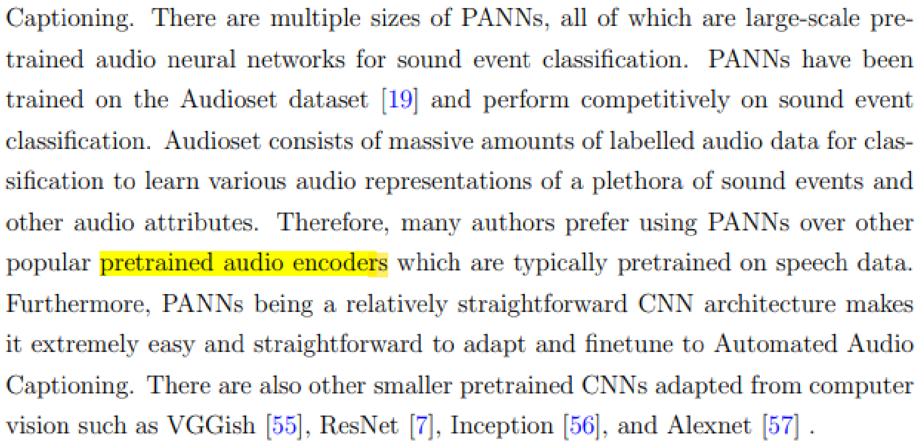
   Response: Thank you for this comment. We have standardized expressions using "low" and "high" to "impulse" and "background"



that occur in shorter impulse events in the foreground. For instance, the model has to detect and describe in the same caption the impulsive audio events such as ”car honking” which occur within a short time frame, and also background events such as ”people having a conversation” that happen over a longer period. Impulse events have a very short duration and and almost always overlap with background events. On the other hand, background events occur over a relatively longer period of time and often form a majority of the audio clip. This overlap between background and impulse events underscores the necessity for the AAC model to recognize these concurrent events, and refrain from conflating them into a singular event.

6. **Examiner's comment:** Page 20: It is not clear if "pretrained audio encoder" refers to one or multiple encoders. In either case, the language should be corrected (either by adding "a" or "the" or changing to plural").

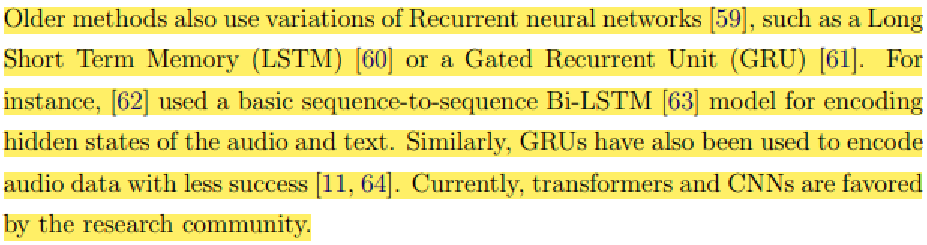
   Response: Thank you for this comment. We have fixed the language by changing the expression to plural.



Captioning. There are multiple sizes of PANNs, all of which are large-scale pretrained audio neural networks for sound event classification. PANNs have been trained on the Audioset dataset [19] and perform competitively on sound event classification. Audioset consists of massive amounts of labelled audio data for classification to learn various audio representations of a plethora of sound events and other audio attributes. Therefore, many authors prefer using PANNs over other popular pretrained audio encoders which are typically pretrained on speech data. Furthermore, PANNs being a relatively straightforward CNN architecture makes it extremely easy and straightforward to adapt and finetune to Automated Audio Captioning. There are also other smaller pretrained CNNs adapted from computer vision such as VGGish [55], ResNet [7], Inception [56], and Alexnet [57] .

7. **Examiner's comment:** Page 20 refers to LSTMs, GRUs, and RNNs as alternative encoder components. However, LSTMs and RNNs are specific types of RNNs, so the description does not give very accurate information about their relationship.

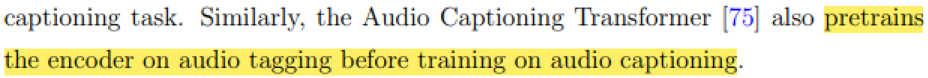
   Response: Thank you for this comment. We have revised the paragraph to indicate that LSTMs and GRUs are variants of RNNs.



Older methods also use variations of Recurrent neural networks [59], such as a Long Short Term Memory (LSTM) [60] or a Gated Recurrent Unit (GRU) [61]. For instance, [62] used a basic sequence-to-sequence Bi-LSTM [63] model for encoding hidden states of the audio and text. Similarly, GRUs have also been used to encode audio data with less success [11, 64]. Currently, transformers and CNNs are favored by the research community.

8. **Examiner's comment:** Page 25 has an expression "pretrains the encoder on audio tagging before training on audio tagging" where I assume the latter "tagging" should be "captioning".

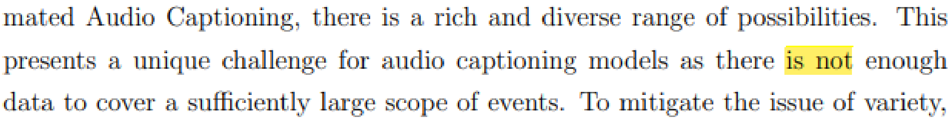
   Response: Thank you for this comment. We have fixed this mistake.



captioning task. Similarly, the Audio Captioning Transformer [75] also pretrains the encoder on audio tagging before training on audio captioning.

9. **Examiner's comment:** Page 27: scientific text should not use contractions such as "isn't".

   Response: Thank you for this comment. We have fixed the language.

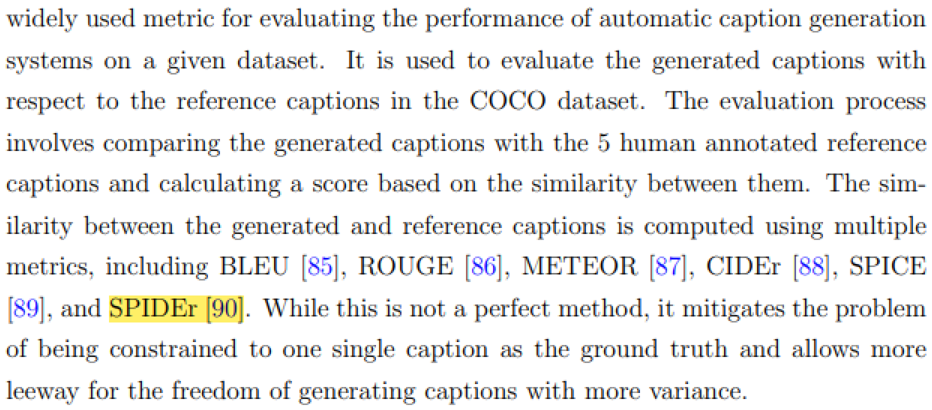


mated Audio Captioning, there is a rich and diverse range of possibilities. This presents a unique challenge for audio captioning models as there is not enough data to cover a sufficiently large scope of events. To mitigate the issue of variety,

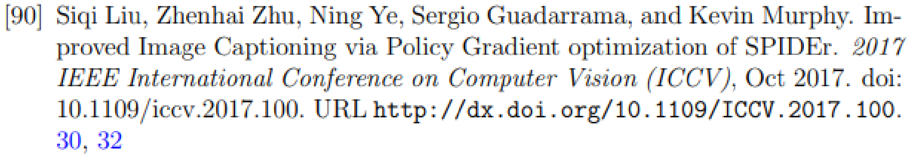
10. **Examiner's comment:** Page 32: text "SPIDEr [90] (portmanteau of SPICE and CIDEr) is a linear combination of SPICE and CIDEr, and was optimized by using a policy gradient [92] method" gives the impression that the metric was optimized using the policy gradient method. However, reference [90] seems to be about optimizing a machine learning model (given a fixed metric), so the text is misleading. 4 (7)

    Response: Thank you for this comment. We would like to clarify that the reference is about using a policy gradient method to optimize SPIDEr. Please refer to snippets taken from the reference below.

Reference in question [90]:

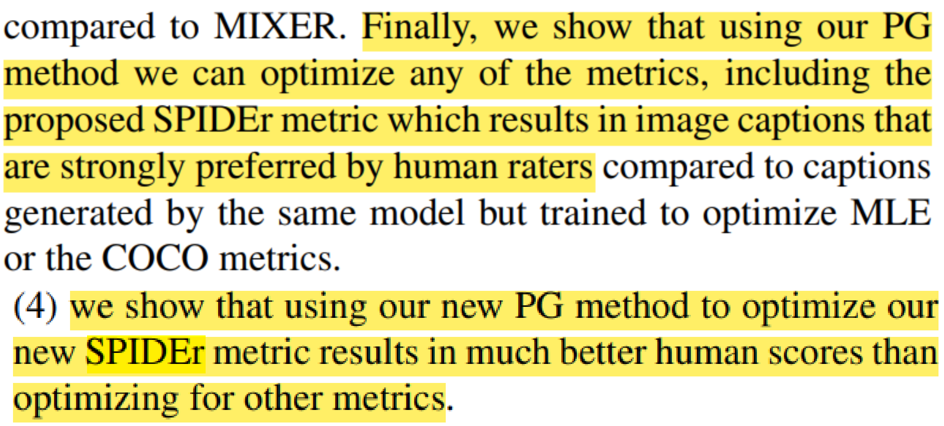


widely used metric for evaluating the performance of automatic caption generation systems on a given dataset. It is used to evaluate the generated captions with respect to the reference captions in the COCO dataset. The evaluation process involves comparing the generated captions with the 5 human annotated reference captions and calculating a score based on the similarity between them. The similarity between the generated and reference captions is computed using multiple metrics, including BLEU [85], ROUGE [86], METEOR [87], CIDEr [88], SPICE [89], and SPIDEr [90]. While this is not a perfect method, it mitigates the problem of being constrained to one single caption as the ground truth and allows more leeway for the freedom of generating captions with more variance.



[90] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved Image Captioning via Policy Gradient optimization of SPIDEr. *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. doi: 10.1109/iccv.2017.100. URL `http://dx.doi.org/10.1109/ICCV.2017.100`. 30, 32

Taken from the reference [90]

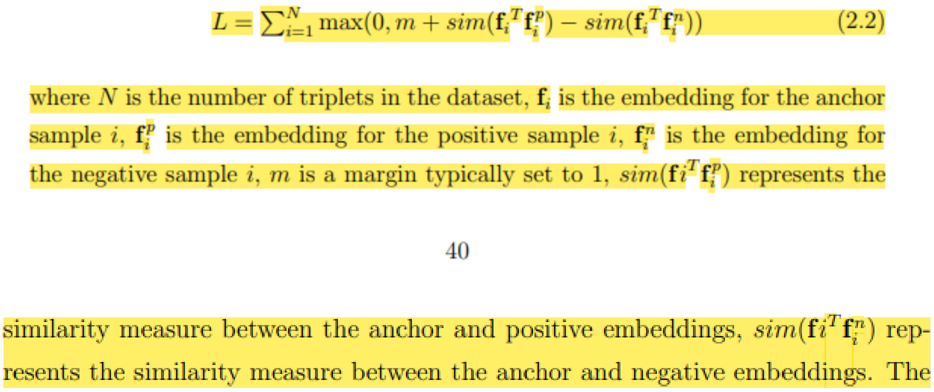


compared to MIXER. Finally, we show that using our PG method we can optimize any of the metrics, including the proposed SPIDEr metric which results in image captions that are strongly preferred by human raters compared to captions generated by the same model but trained to optimize MLE or the COCO metrics.

(4) we show that using our new PG method to optimize our new SPIDEr metric results in much better human scores than optimizing for other metrics.

11. **Examiner's comment:** Page 40: variables fpi and fni use inconsistent formatting; in some expressions, i, n, and p are superscripts of f, but in some expressions they are not. In general, the mathematical notations should be consistent.

    Response: Thank you for this comment. We have standardized the mathematical notations.



$$L = \sum_{i=1}^{N} \max(0, m + sim(\mathbf{f}_i{}^T \mathbf{f}_i^p) - sim(\mathbf{f}_i{}^T \mathbf{f}_i^n)) \qquad (2.2)$$

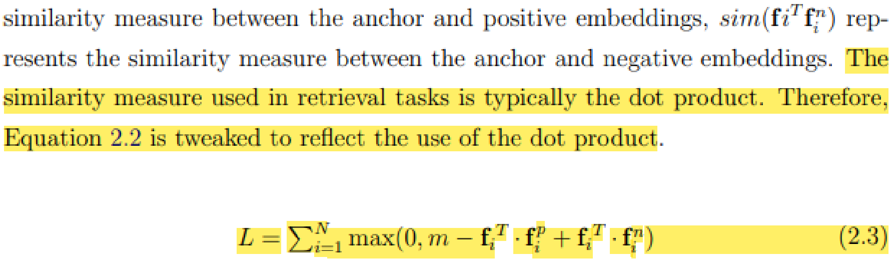
where N is the number of triplets in the dataset, $\mathbf{f}_i$ is the embedding for the anchor sample i, $\mathbf{f}_i^p$ is the embedding for the positive sample i, $\mathbf{f}_i^n$ is the embedding for the negative sample i, m is a margin typically set to 1, $sim(\mathbf{f}i^T \mathbf{f}_i^p)$ represents the

40

similarity measure between the anchor and positive embeddings, $sim(\mathbf{f}i^T \mathbf{f}_i^n)$ represents the similarity measure between the anchor and negative embeddings. The

12. **Examiner's comment:** Page 40: Eq. (2.2) defining the triplet loss uses dot products as similarity measures. However, this definition of the triples loss would lead to minimizing the similarity of positive samples and maximizing the similarity of dissimilar ones, so there is a sign error in the formula.

    Response: Thank you for this comment. The original triplet ranking loss shown is the standard triplet ranking loss formula. We have also added Equation 2.3 to reflect the use of the dot product in the triplet ranking loss.

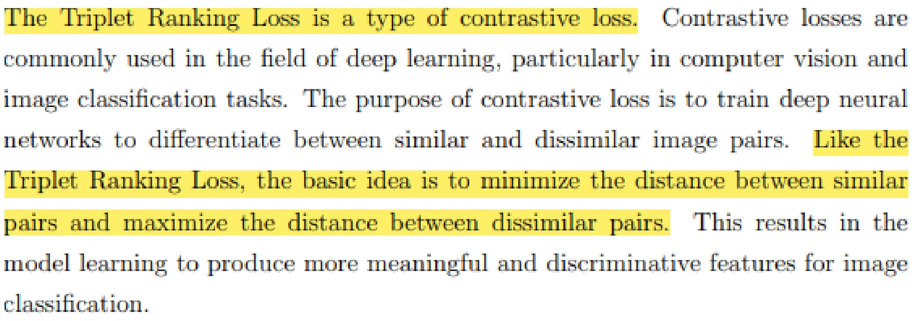


similarity measure between the anchor and positive embeddings, $sim(\mathbf{f}i^T \mathbf{f}_i^n)$ represents the similarity measure between the anchor and negative embeddings. The similarity measure used in retrieval tasks is typically the dot product. Therefore, Equation 2.2 is tweaked to reflect the use of the dot product.

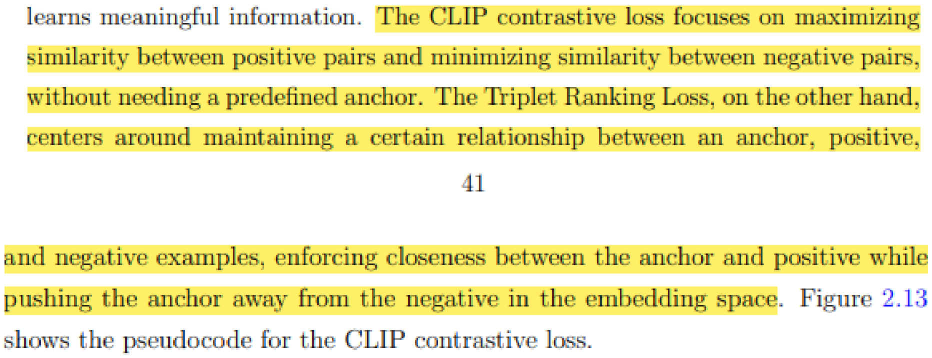
$$L = \sum_{i=1}^{N} \max(0, m - \mathbf{f}_i{}^T \cdot \mathbf{f}_i^p + \mathbf{f}_i{}^T \cdot \mathbf{f}_i^n) \qquad (2.3)$$

13. **Examiner's comment:** Page 41: It is described that a contrastive loss is used in the thesis, and that contrastive loss has not previously been used in language based audio retrieval. However, the triplet loss which has been extensively used on the topic is also a contrastive loss, so this description is misleading. In general, the relationship between the triplet loss and the other contrastive loss that is used should be made more clear to understand what value the other contrastive loss adds to the triplet loss.

    Response: Thank you for this comment. We have revised the section to clarify the difference and highlight the value that the contrastive loss provides.

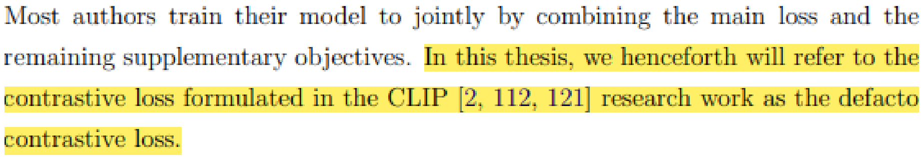


The Triplet Ranking Loss is a type of contrastive loss. Contrastive losses are commonly used in the field of deep learning, particularly in computer vision and image classification tasks. The purpose of contrastive loss is to train deep neural networks to differentiate between similar and dissimilar image pairs. Like the Triplet Ranking Loss, the basic idea is to minimize the distance between similar pairs and maximize the distance between dissimilar pairs. This results in the model learning to produce more meaningful and discriminative features for image classification.



learns meaningful information. The CLIP contrastive loss focuses on maximizing similarity between positive pairs and minimizing similarity between negative pairs, without needing a predefined anchor. The Triplet Ranking Loss, on the other hand, centers around maintaining a certain relationship between an anchor, positive, and negative examples, enforcing closeness between the anchor and positive while pushing the anchor away from the negative in the embedding space. Figure 2.13 shows the pseudocode for the CLIP contrastive loss.

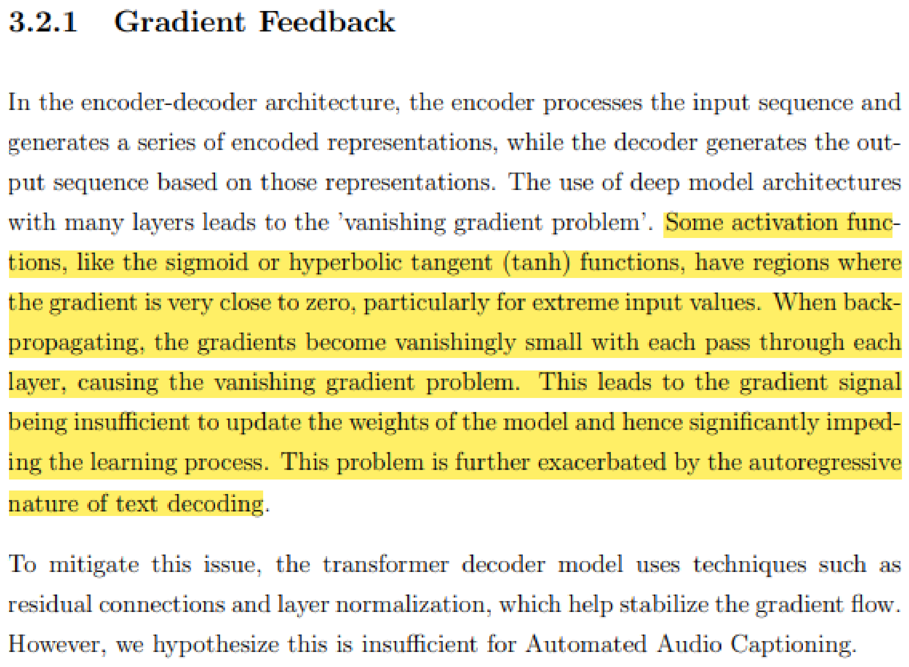
We have also highlighted that the contrastive loss referred to the thesis is the CLIP contrastive loss to avoid confusion.



Most authors train their model to jointly by combining the main loss and the remaining supplementary objectives. In this thesis, we henceforth will refer to the contrastive loss formulated in the CLIP [2, 112, 121] research work as the defacto contrastive loss.

14. **Examiner’s comment:** Page 53: In section 3.2.1, the vanishing gradient problem is described to originate from the use of attention mechanism. This is misleading, since the attention as such does not cause the vanishing gradient problem. The problem originates more from the autoregressive sequence models where all the previous time steps are used as an input in the decoder.

    Response: Thank you for this comment. We have rephrased misleading sentences to reflect that the vanishing gradient problem comes from depth of the model when back-propagating.



3.2.1 Gradient Feedback

In the encoder-decoder architecture, the encoder processes the input sequence and generates a series of encoded representations, while the decoder generates the output sequence based on those representations. The use of deep model architectures with many layers leads to the 'vanishing gradient problem'. Some activation functions, like the sigmoid or hyperbolic tangent (tanh) functions, have regions where the gradient is very close to zero, particularly for extreme input values. When back-propagating, the gradients become vanishingly small with each pass through each layer, causing the vanishing gradient problem. This leads to the gradient signal being insufficient to update the weights of the model and hence significantly impeding the learning process. This problem is further exacerbated by the autoregressive nature of text decoding.

To mitigate this issue, the transformer decoder model uses techniques such as residual connections and layer normalization, which help stabilize the gradient flow. However, we hypothesize this is insufficient for Automated Audio Captioning.

15. **Examiner's comment:** Pages 55-60: the dimensionality of data representation at different processing steps is not given, and specifically there is no information what is the dimensionality of the data that is inputted to the transformer. This makes it difficult to understand the functioning and role of the transformer encoder properly. Based on the illustration of figure 3.2, the output of the CNN encoders is a 527-length vector, but in this case there would be no need for any sequence model like a transformer, which is confusing. Since on page 55 it is described that testing transformer encoder fully in audio captioning is one of the goals of this chapter, the processing steps and data in each of them should be described in sufficient detail to allow understanding how the processing is exactly done.

    Response: Thank you for this comment. In the caption of Figure 3.2, we have added clarification that the features before the final fully connected layers are used as embeddings. We have also updated Figure 3.3 to fully reflect the dimension sizes of the embeddings at each intermediate layer. The full list of hyperparameters is shown in Table 3.1 and Table 3.2.

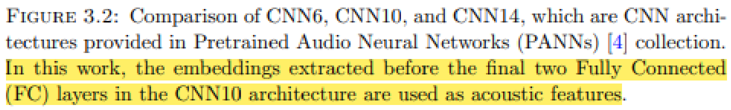


FIGURE 3.2: Comparison of CNN6, CNN10, and CNN14, which are CNN architectures provided in Pretrained Audio Neural Networks (PANNs) [4] collection. In this work, the embeddings extracted before the final two Fully Connected (FC) layers in the CNN10 architecture are used as acoustic features.

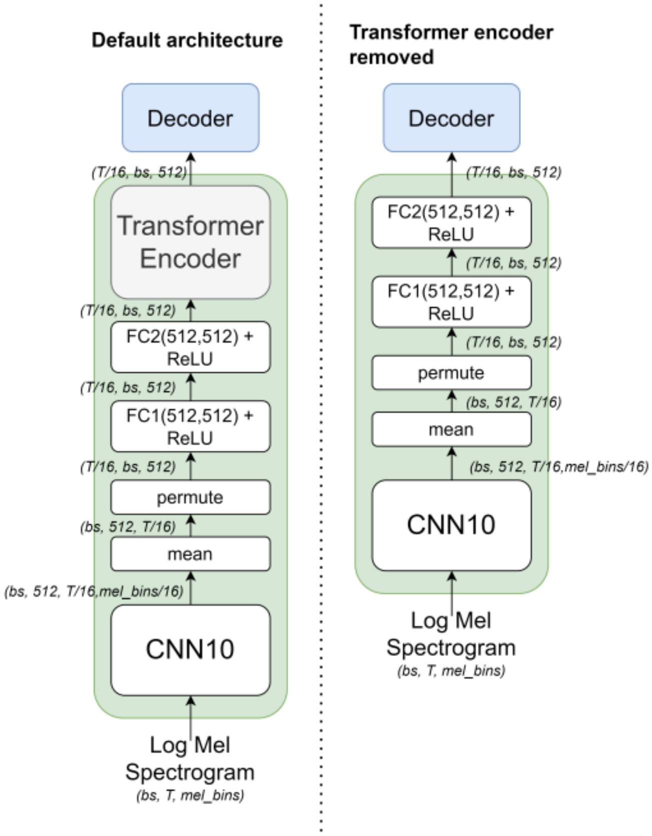


FIGURE 3.3: Left: default audio encoder architecture with CNN10 and the appended transformer encoder. Right: audio encoder architecture with only CNN10. The output acoustic embedding of the whole encoder is passed to the decoder.

Parameter	Value
sampling rate	44100
length of the FFT window	1024
hop size	512
number of Mel bands	64
window function	hann
center	Yes
htk	No
power	1.0 (energy)
norm	1.0

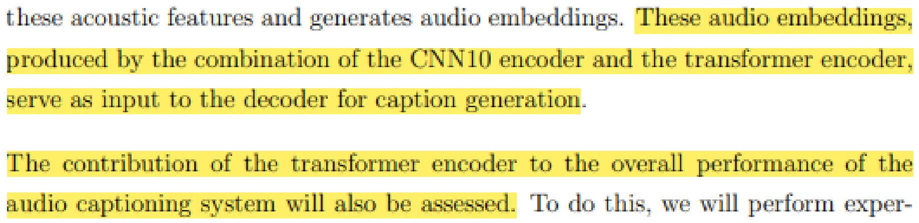
TABLE 3.1: Parameters used in librosa to preprocess each audio clip into log mel spectrograms. "htk" being false refers to using the Auditory toolbox [5] implementation as opposed to the Hidden Markov Toolkit (HTK) implementation [6] to preprocess the audio waveform into mel spectrograms.

Hyperparameter	**Value**
batch size	64
number of attention heads	4
hidden size	192
number of transformer layers	2
number of dictionary tokens	4368
gradient clipping	2.5
learning rate	0.0003
inference beam width	4

Table 3.2: Hyperparameters

16. **Examiner's comment:** Page 60: There is two incomplete sentences "which uses them to generate capt" and "To assess the contribution of the transformer encoder to the overall performance of the audio captioning system."

    Response: Thank you for this comment. We have fixed the typos.

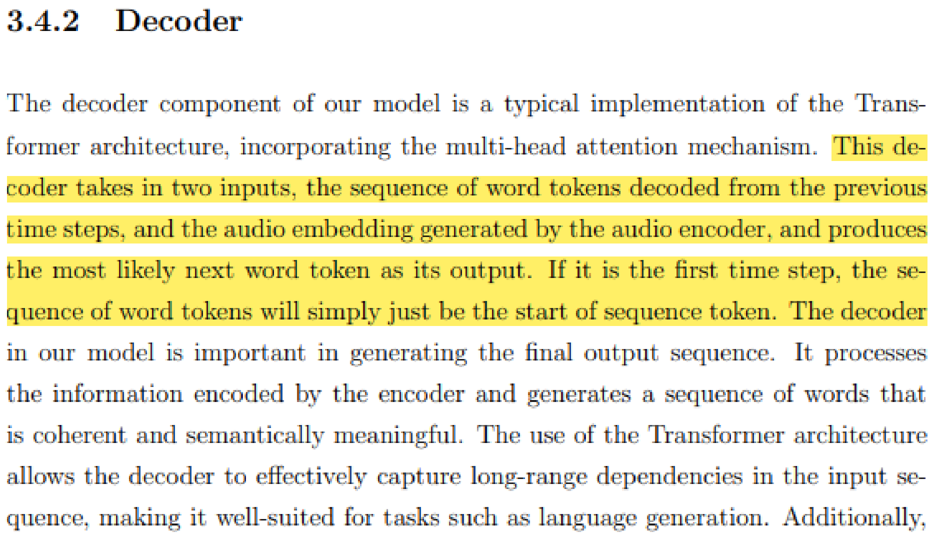


these acoustic features and generates audio embeddings. These audio embeddings, produced by the combination of the CNN10 encoder and the transformer encoder, serve as input to the decoder for caption generation.

The contribution of the transformer encoder to the overall performance of the audio captioning system will also be assessed. To do this, we will perform exper-

17. **Examiner's comment:** Page 60: It is described that the input of the decoder includes a sequence of word tokens. However, it is not clear what word tokens these are. I assume they are the previously outputted words, but this is not explained properly. In general, the functioning of the de coder should be explained at sufficient detail to allow reproducing the results.

    Response: Thank you for this comment. We have revised the section to delineate the process of decoding.

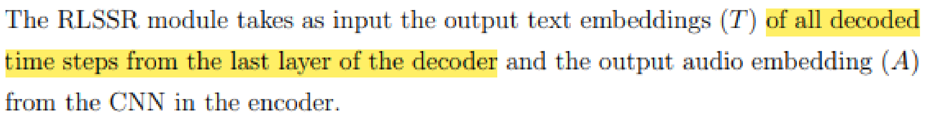


3.4.2 Decoder

The decoder component of our model is a typical implementation of the Transformer architecture, incorporating the multi-head attention mechanism. This decoder takes in two inputs, the sequence of word tokens decoded from the previous time steps, and the audio embedding generated by the audio encoder, and produces the most likely next word token as its output. If it is the first time step, the sequence of word tokens will simply just be the start of sequence token. The decoder in our model is important in generating the final output sequence. It processes the information encoded by the encoder and generates a sequence of words that is coherent and semantically meaningful. The use of the Transformer architecture allows the decoder to effectively capture long-range dependencies in the input sequence, making it well-suited for tasks such as language generation. Additionally,

18. **Examiner's comment:** Page 61: The text describes that "The RLSSR module takes as input the output text embedding (T) from the last layer of the decoder". However, it is not clear if this is the sequence of embeddings or an embedding of a specific time step.

    Response: Thank you for this comment. We have added clarification that it is of all decoded time steps.

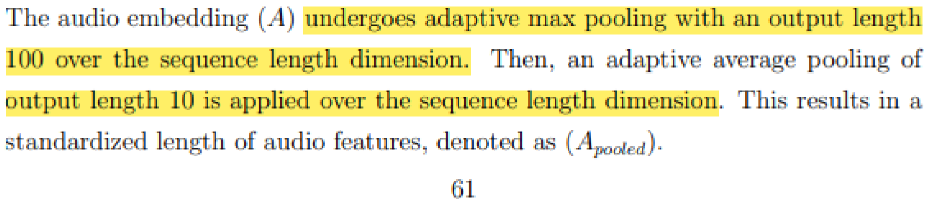


The RLSSR module takes as input the output text embeddings (T) of all decoded time steps from the last layer of the decoder and the output audio embedding (A) from the CNN in the encoder.

\

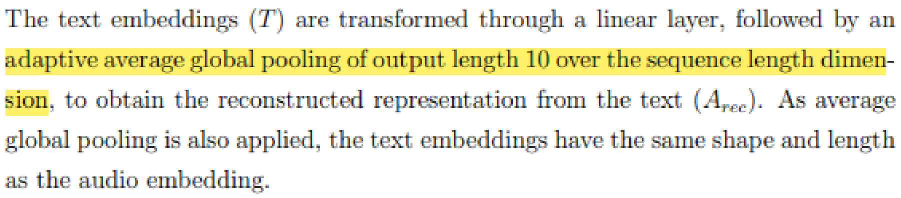
19. **Examiner's comment:** Page 61 refers to "average max pooling" and "average global pooling" operations, but it has not been defined how these exactly operate (which operations are done over what dimensions)

    Response: Thank you for this comment. We have clarified the paragraph to reflect that the average max pooling and average global pooling operations are applied over the sequence dimension in order to standardize the length of the sequence of both audio and text embeddings.



The audio embedding (A) undergoes adaptive max pooling with an output length 100 over the sequence length dimension. Then, an adaptive average pooling of output length 10 is applied over the sequence length dimension. This results in a standardized length of audio features, denoted as (A_{pooled}).

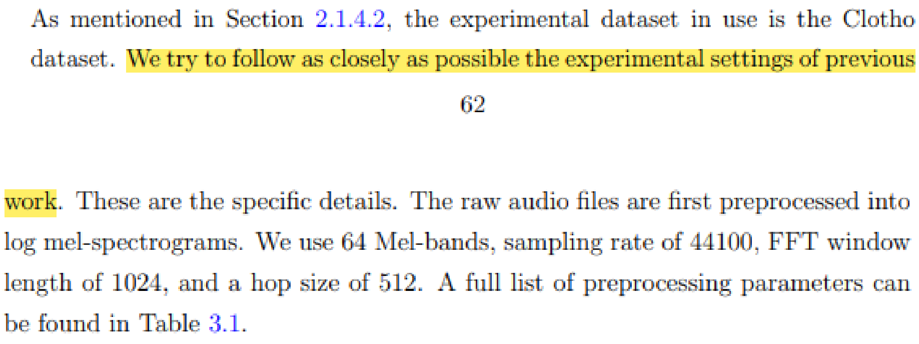
61



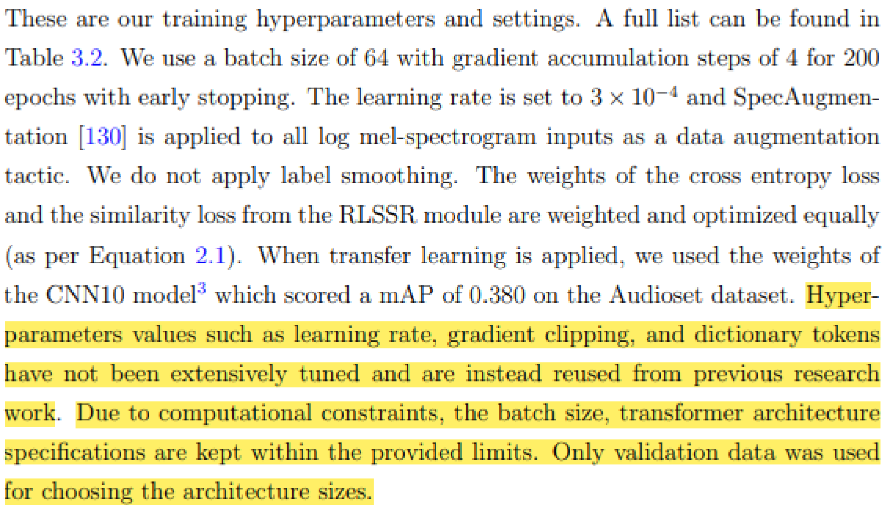
The text embeddings (T) are transformed through a linear layer, followed by an adaptive average global pooling of output length 10 over the sequence length dimension, to obtain the reconstructed representation from the text (A_{rec}). As average global pooling is also applied, the text embeddings have the same shape and length as the audio embedding.

20. **Examiner's comment:** Pages 63 and 77 describe the hyperparameters used in the experiments. It has not been explained what criteria has been used to choose their values. Particularly it should be explained whether validation or test data is used for choosing or optimizing the hyperparameter values.

    Response: Thank you for this comment. We have added additional statements to clarify that the values used are similar with previous captioning research. We did not perform extensive hyperparameter tuning.



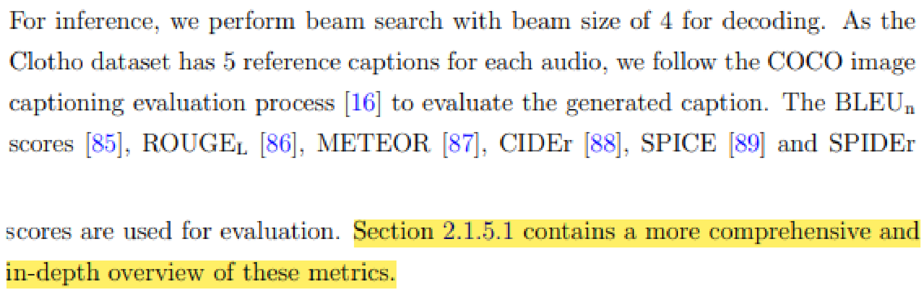
As mentioned in Section 2.1.4.2, the experimental dataset in use is the Clotho dataset. We try to follow as closely as possible the experimental settings of previous work. These are the specific details. The raw audio files are first preprocessed into log mel-spectrograms. We use 64 Mel-bands, sampling rate of 44100, FFT window length of 1024, and a hop size of 512. A full list of preprocessing parameters can be found in Table 3.1.



These are our training hyperparameters and settings. A full list can be found in Table 3.2. We use a batch size of 64 with gradient accumulation steps of 4 for 200 epochs with early stopping. The learning rate is set to 3×10^{-4} and SpecAugmentation [130] is applied to all log mel-spectrogram inputs as a data augmentation tactic. We do not apply label smoothing. The weights of the cross entropy loss and the similarity loss from the RLSSR module are weighted and optimized equally (as per Equation 2.1). When transfer learning is applied, we used the weights of the CNN10 model[3] which scored a mAP of 0.380 on the Audioset dataset. Hyperparameters values such as learning rate, gradient clipping, and dictionary tokens have not been extensively tuned and are instead reused from previous research work. Due to computational constraints, the batch size, transformer architecture specifications are kept within the provided limits. Only validation data was used for choosing the architecture sizes.

21. **Examiner's comment:** Page 62 describes the procedures for calculating the evaluation metrics. This is redundant with the explanation in Section 2.1.5.

    Response: Thank you for this comment. We have shortened the paragraph and made a reference to Section 2.1.5.

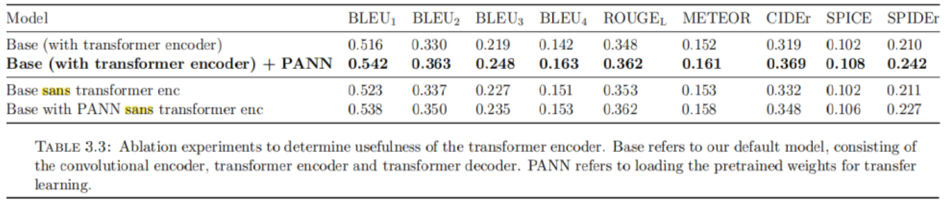


For inference, we perform beam search with beam size of 4 for decoding. As the Clotho dataset has 5 reference captions for each audio, we follow the COCO image captioning evaluation process [16] to evaluate the generated caption. The BLEU$_n$ scores [85], ROUGE$_L$ [86], METEOR [87], CIDEr [88], SPICE [89] and SPIDEr

scores are used for evaluation. Section 2.1.5.1 contains a more comprehensive and in-depth overview of these metrics.

22. **Examiner's comment:** Page 64, Table 3.3 included method "Base – transformer enc". It is not fully clear whether this is a model with or without the transformer encoder.

    Response: Thank you for this comment. We have replaced "-" with "sans" to indicate that the model has its transformer encoder removed.

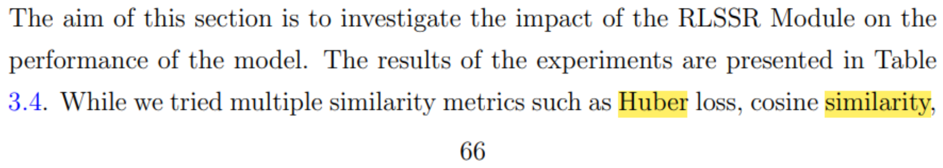


Model	BLEU$_1$	BLEU$_2$	BLEU$_3$	BLEU$_4$	ROUGE$_L$	METEOR	CIDEr	SPICE	SPIDEr
Base (with transformer encoder)	0.516	0.330	0.219	0.142	0.348	0.152	0.319	0.102	0.210
Base (with transformer encoder) + PANN	**0.542**	**0.363**	**0.248**	**0.163**	**0.362**	**0.161**	**0.369**	**0.108**	**0.242**
Base sans transformer enc	0.523	0.337	0.227	0.151	0.353	0.153	0.332	0.102	0.211
Base with PANN sans transformer enc	0.538	0.350	0.235	0.153	0.362	0.158	0.348	0.106	0.227

TABLE 3.3: Ablation experiments to determine usefulness of the transformer encoder. Base refers to our default model, consisting of the convolutional encoder, transformer encoder and transformer decoder. PANN refers to loading the pretrained weights for transfer learning.

23. **Examiner's comment:** Page 65: "huber" should be capitalized, typo "simliarity"

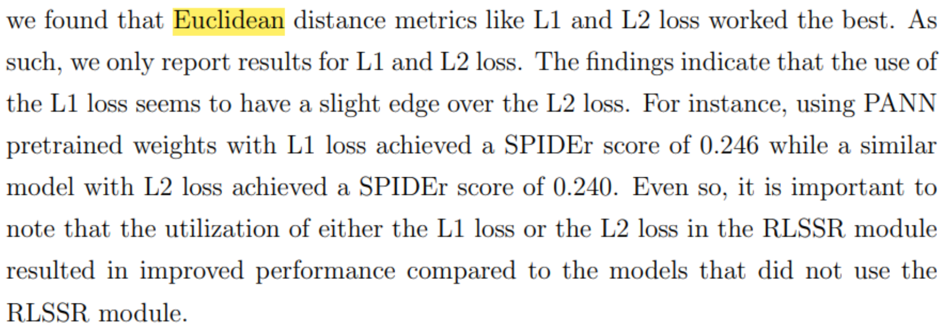
    Response: Thank you for this comment. We have fixed the errors.



The aim of this section is to investigate the impact of the RLSSR Module on the performance of the model. The results of the experiments are presented in Table 3.4. While we tried multiple similarity metrics such as Huber loss, cosine similarity,

24. **Examiner's comment:** Page 66: "euclidean" should be capitalized

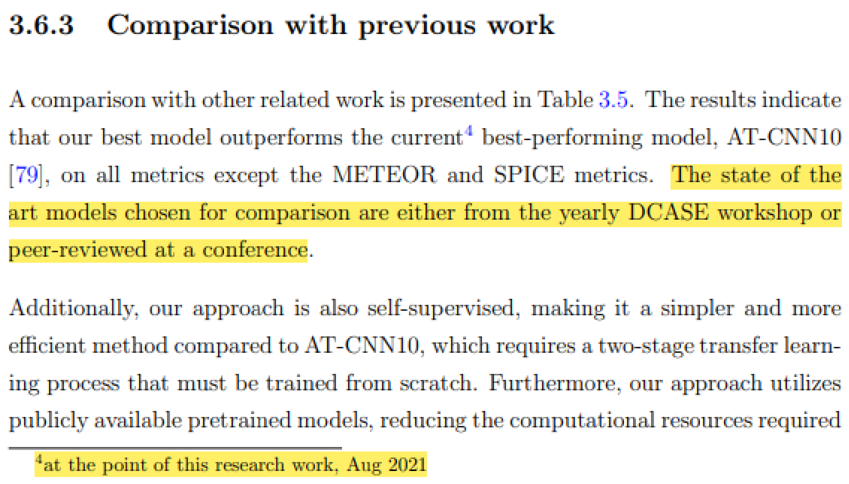
Response: Thank you for this comment. We have fixed the capitalization.



we found that Euclidean distance metrics like L1 and L2 loss worked the best. As such, we only report results for L1 and L2 loss. The findings indicate that the use of the L1 loss seems to have a slight edge over the L2 loss. For instance, using PANN pretrained weights with L1 loss achieved a SPIDEr score of 0.246 while a similar model with L2 loss achieved a SPIDEr score of 0.240. Even so, it is important to note that the utilization of either the L1 loss or the L2 loss in the RLSSR module resulted in improved performance compared to the models that did not use the RLSSR module.

25. **Examiner's comment:** Page 66: The best developed method is described to outperform the best related model. It is not fully clear how the reference methods are chosen. For audio captioning there exists several models that produce clearly better results than the ones presented in the thesis. There is no need that the proposed method outperform those, but it will be good to discuss how the proposed methods compare to the state of the art, and what are the factors affecting the reasons why there are methods which performance is clearly better. Table 3.5 that presents the results of the methods refers to these other models as "state of the art". However, there existing several methods that give significantly better results than the models presented in the Table, so it should be clarified in what terms the chosen methods are chosen state of the art.

Response: Thank you for this comment. We have added clarification that the comparisons were made at the point of research and writing. The models are either from the DCASE workshop or submitted to a peer-reviewed conference.



3.6.3 Comparison with previous work

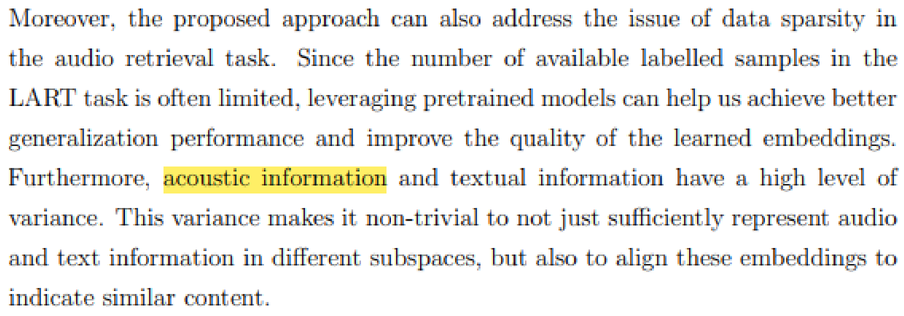
A comparison with other related work is presented in Table 3.5. The results indicate that our best model outperforms the current[4] best-performing model, AT-CNN10 [79], on all metrics except the METEOR and SPICE metrics. The state of the art models chosen for comparison are either from the yearly DCASE workshop or peer-reviewed at a conference.

Additionally, our approach is also self-supervised, making it a simpler and more efficient method compared to AT-CNN10, which requires a two-stage transfer learning process that must be trained from scratch. Furthermore, our approach utilizes publicly available pretrained models, reducing the computational resources required

[4] at the point of this research work, Aug 2021

26. **Examiner’s comment:** Page 72: “Acoustic information” should be lowercase.

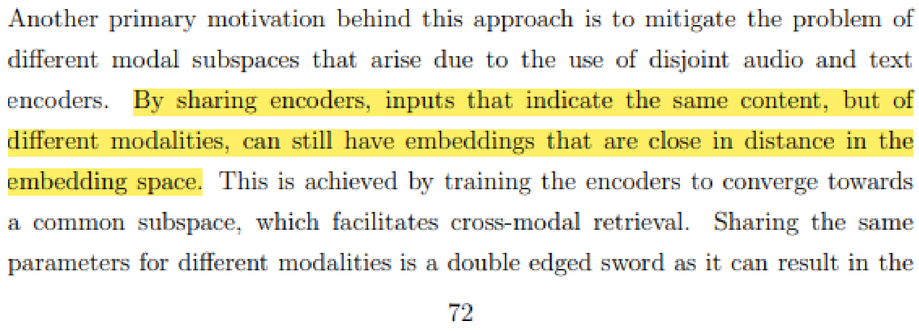
    Response: Thank you for this comment. We have fixed the capitalization.



Moreover, the proposed approach can also address the issue of data sparsity in the audio retrieval task. Since the number of available labelled samples in the LART task is often limited, leveraging pretrained models can help us achieve better generalization performance and improve the quality of the learned embeddings. Furthermore, acoustic information and textual information have a high level of variance. This variance makes it non-trivial to not just sufficiently represent audio and text information in different subspaces, but also to align these embeddings to indicate similar content.

27. **Examiner’s comment:** Page 72: The expression “minimally close in distance” is confusing. It can be understood as the closeness being minimal (e.g., things are distant), even though I think the point is that the distance is minimal. This could be rephrased to avoid ambiguity.

    Response: Thank you for this comment. We have rephrased for the statement for clarity.

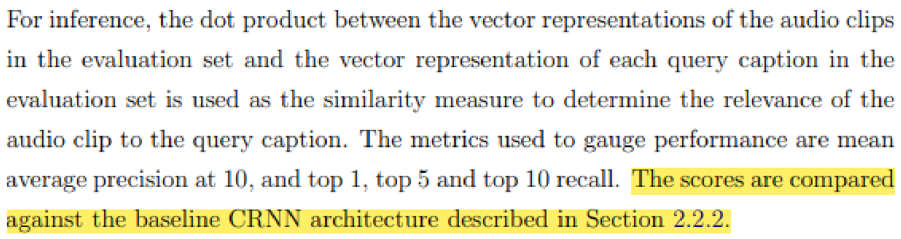


Another primary motivation behind this approach is to mitigate the problem of different modal subspaces that arise due to the use of disjoint audio and text encoders. By sharing encoders, inputs that indicate the same content, but of different modalities, can still have embeddings that are close in distance in the embedding space. This is achieved by training the encoders to converge towards a common subspace, which facilitates cross-modal retrieval. Sharing the same parameters for different modalities is a double edged sword as it can result in the

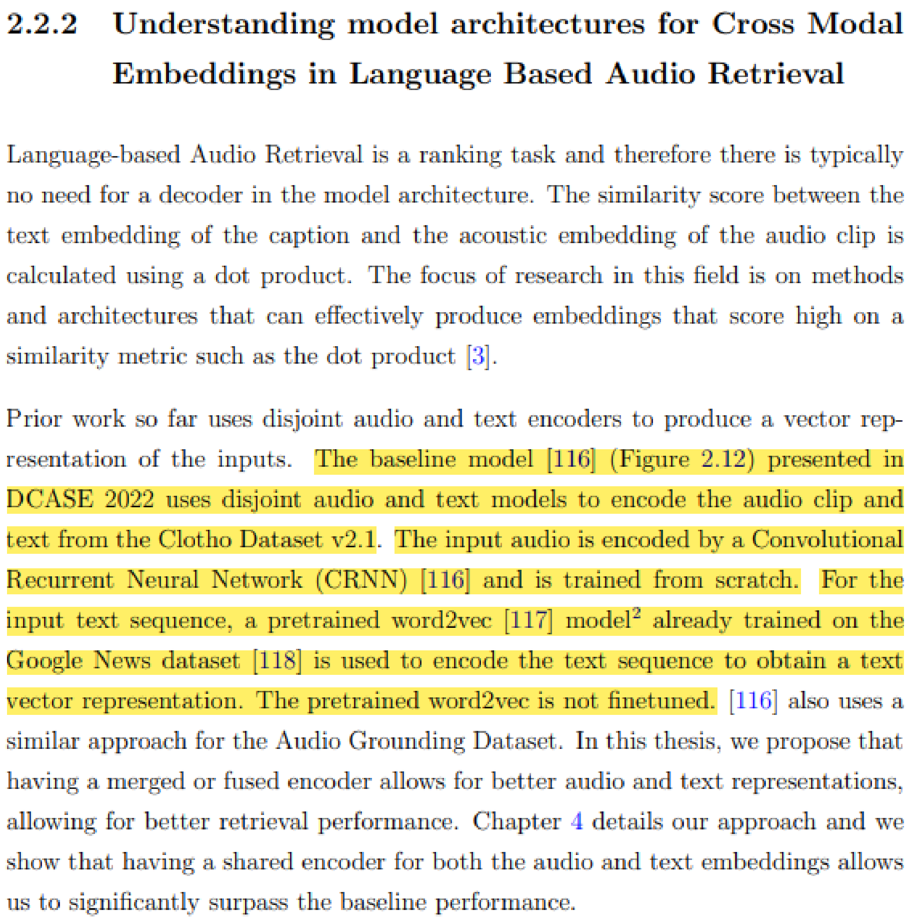
72

28. **Examiner's comment:** Page 73: There is no information about the baseline CRNN model architecture.

    Response: Thank you for this comment. The baseline CRNN model was described in Section 2.2.2. We have added a reference to that section for clarity. In addition, Figure 4.1 contains an overview of the baseline system.



For inference, the dot product between the vector representations of the audio clips in the evaluation set and the vector representation of each query caption in the evaluation set is used as the similarity measure to determine the relevance of the audio clip to the query caption. The metrics used to gauge performance are mean average precision at 10, and top 1, top 5 and top 10 recall. The scores are compared against the baseline CRNN architecture described in Section 2.2.2.



2.2.2 Understanding model architectures for Cross Modal Embeddings in Language Based Audio Retrieval

Language-based Audio Retrieval is a ranking task and therefore there is typically no need for a decoder in the model architecture. The similarity score between the text embedding of the caption and the acoustic embedding of the audio clip is calculated using a dot product. The focus of research in this field is on methods and architectures that can effectively produce embeddings that score high on a similarity metric such as the dot product [3].

Prior work so far uses disjoint audio and text encoders to produce a vector representation of the inputs. The baseline model [116] (Figure 2.12) presented in DCASE 2022 uses disjoint audio and text models to encode the audio clip and text from the Clotho Dataset v2.1. The input audio is encoded by a Convolutional Recurrent Neural Network (CRNN) [116] and is trained from scratch. For the input text sequence, a pretrained word2vec [117] model[2] already trained on the Google News dataset [118] is used to encode the text sequence to obtain a text vector representation. The pretrained word2vec is not finetuned. [116] also uses a similar approach for the Audio Grounding Dataset. In this thesis, we propose that having a merged or fused encoder allows for better audio and text representations, allowing for better retrieval performance. Chapter 4 details our approach and we show that having a shared encoder for both the audio and text embeddings allows us to significantly surpass the baseline performance.

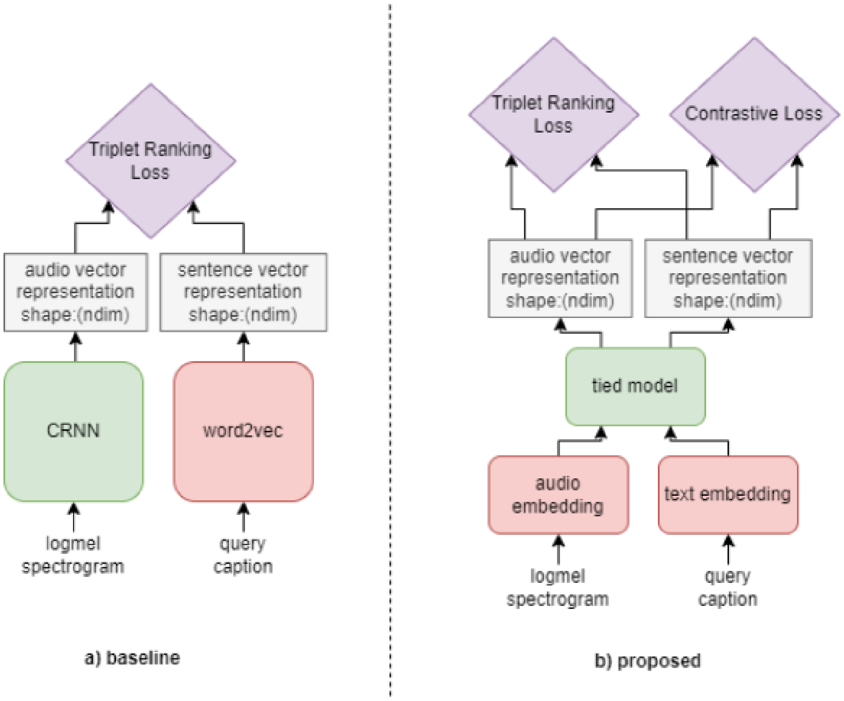
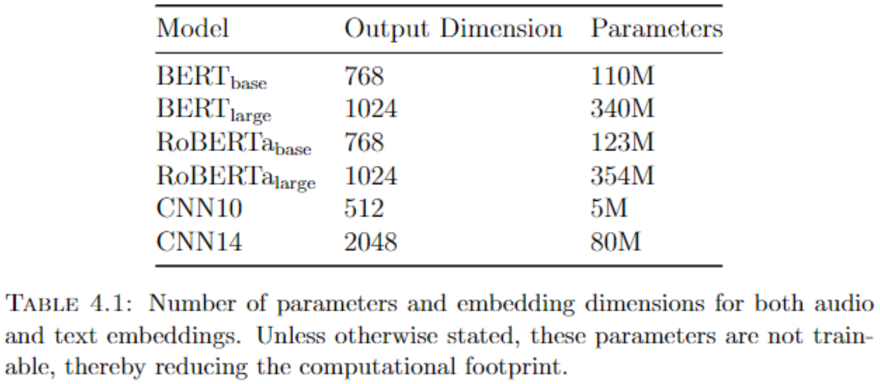


FIGURE 4.1: **a)** Baseline system. A CRNN is trained using the triplet ranking loss for the audio encoder while a word2vec model pretrained on Google News is used without any finetuning. **b)** Proposed system. Both audio embeddings and text embeddings are used with frozen weights without any finetuning. We use CNN10, CNN14 for the audio embeddings and BERT, RoBERTa for the text embeddings. Both embeddings are passed to the tied layers which are trained on both Triplet Ranking Loss and Contrastive Loss. Shaded red boxes in the figure refers to models with frozen parameters (not finetuned) while green boxes refers to layers/models with trainable parameters.

29. **Examiner's comment:** Pages 73-76: the proposed method consists of CNN layers to produce audio embeddings. There is no information about the dimensionality of the embeddings, which makes it difficult to understand the role of the studied transformer or tied layers.

    Response: Thank you for this comment. The dimensionality of the embeddings is dependent on the frozen encoder used in our experiments. We have updated Table 4.1 to reflect the dimensionality of these encoders.

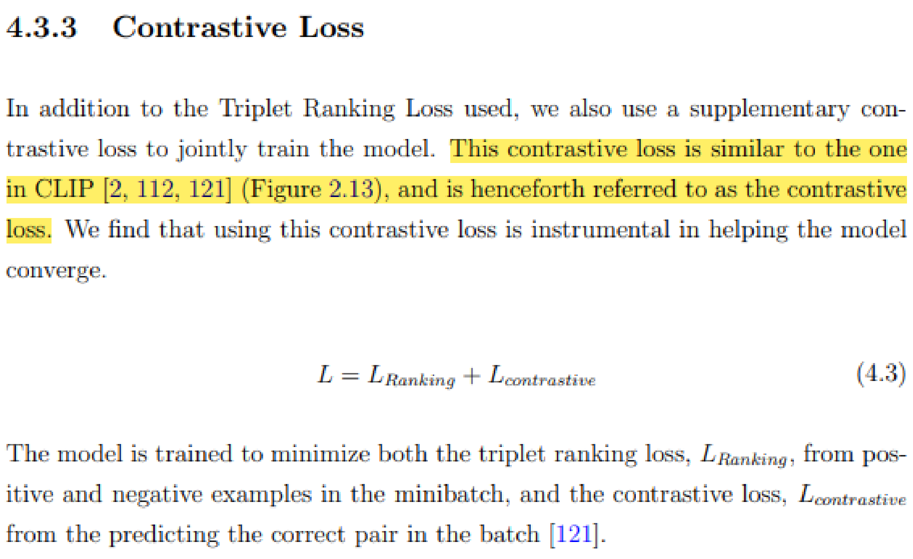


Model	Output Dimension	Parameters
BERT$_{\text{base}}$	768	110M
BERT$_{\text{large}}$	1024	340M
RoBERTa$_{\text{base}}$	768	123M
RoBERTa$_{\text{large}}$	1024	354M
CNN10	512	5M
CNN14	2048	80M

Table 4.1: Number of parameters and embedding dimensions for both audio and text embeddings. Unless otherwise stated, these parameters are not trainable, thereby reducing the computational footprint.

30. **Examiner's comment:** Page 76: The text refers to Section 2.3 for the contrastive loss, but Section 2.3 does not contain information about the loss. It will be good to include a formula for the contrastive loss used, to understand how it differs from the triplet ranking loss that is used as a part of the total loss. The triplet ranking loss is also a contrastive loss, so terming the supplementary contrastive only as a contrastive loss is misleading, and unclear how these two contrastive losses complement each other.

    Response: Thank you for this comment. The contrastive loss has been mentioned a few times in Section 2 (CLIP contrastive loss). Please refer to response 13. In addition, we have added a reiteration to avoid confusion.



4.3.3 Contrastive Loss

In addition to the Triplet Ranking Loss used, we also use a supplementary contrastive loss to jointly train the model. This contrastive loss is similar to the one in CLIP [2, 112, 121] (Figure 2.13), and is henceforth referred to as the contrastive loss. We find that using this contrastive loss is instrumental in helping the model converge.

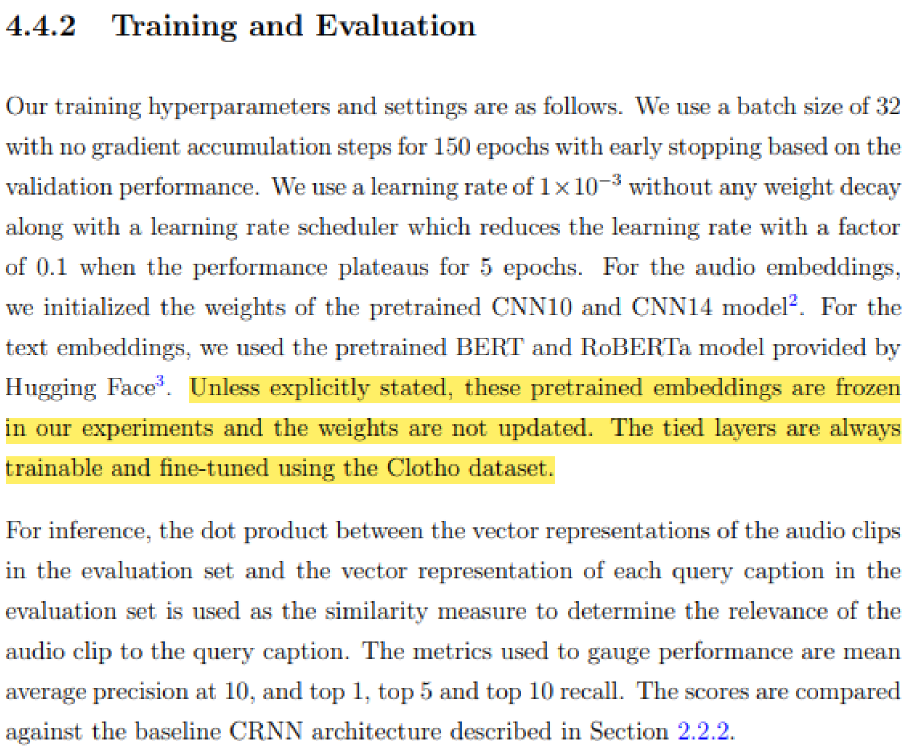
$$L = L_{Ranking} + L_{contrastive} \tag{4.3}$$

The model is trained to minimize both the triplet ranking loss, $L_{Ranking}$, from positive and negative examples in the minibatch, and the contrastive loss, $L_{contrastive}$ from the predicting the correct pair in the batch [121].

31. **Examiner's comment:** Page 77: The model with converging tied layers and contrastive loss is described to perform significantly better than the baseline. However, the encoders of these two models 6 (7) are trained with different data (including the pre-training), so conclusions about the relative performance of different model architectures cannot be drawn. Ideally the models should be compared in a setting where the same training data is used for all the models, so that one can evaluate the effect of the models. Ideally additional experiments should be done with same training data, or if this is not feasible the text should be revised to make it clear that the results cannot be used to draw any conclusions about the effect of the converging tied layers.

    Response: Thank you for this comment. In our case, the use of transfer learning in deep learning is widespread and common. Previous research work also utilizes different common pretrained models like BERT or word2vec which does not use the same training data for pretraining. The basis of our research is to show that by using converging tied layers, it is sufficient to simply finetune these tied layers and freeze the pretrained layers to achieve competitive results, as opposed to having to finetune the whole model.

    It is important to note that the finetuning of the tied layers is all done using the Clotho Dataset. We have also added clarification to indicate that tied layers are always trainable.



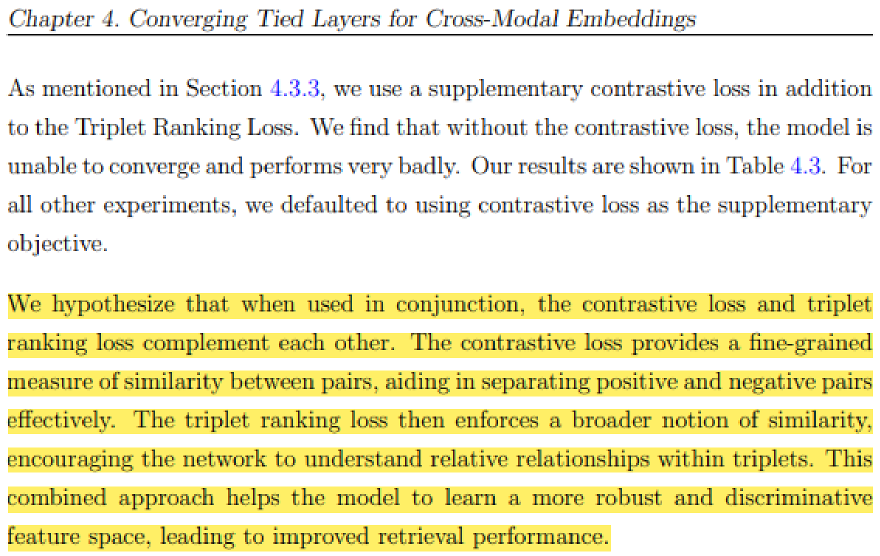
4.4.2 Training and Evaluation

Our training hyperparameters and settings are as follows. We use a batch size of 32 with no gradient accumulation steps for 150 epochs with early stopping based on the validation performance. We use a learning rate of 1×10^{-3} without any weight decay along with a learning rate scheduler which reduces the learning rate with a factor of 0.1 when the performance plateaus for 5 epochs. For the audio embeddings, we initialized the weights of the pretrained CNN10 and CNN14 model[2]. For the text embeddings, we used the pretrained BERT and RoBERTa model provided by Hugging Face[3]. Unless explicitly stated, these pretrained embeddings are frozen in our experiments and the weights are not updated. The tied layers are always trainable and fine-tuned using the Clotho dataset.

For inference, the dot product between the vector representations of the audio clips in the evaluation set and the vector representation of each query caption in the evaluation set is used as the similarity measure to determine the relevance of the audio clip to the query caption. The metrics used to gauge performance are mean average precision at 10, and top 1, top 5 and top 10 recall. The scores are compared against the baseline CRNN architecture described in Section 2.2.2.

32. **Examiner's comment:** Page 78: Table 4.3 reports the results of models trained without the supplementary contrastive loss. However, the CRNN results in the table seem to be the same as the CRNN results in Table 4.2, where the supplementary contrastive loss is presumably included. This is not fully clear since this is not explicitly described, and it is not clear whether the CRNN model in Table 4.3 uses the supplementary contrastive loss or not. It would be good to discuss reasons why the lack of supplementary contrastive make the developed models fail. The caption refers to results "without contrastive loss", but presumably these still include the triplet ranking loss which is also a contrastive loss, so the description is misleading.

    Response: Thank you for this comment. Please refer to related response 13 and 30. The referred contrastive loss is the CLIP contrastive loss. We have added additional analysis on how the contrastive loss complements the triplet ranking loss.



Chapter 4. Converging Tied Layers for Cross-Modal Embeddings

As mentioned in Section 4.3.3, we use a supplementary contrastive loss in addition to the Triplet Ranking Loss. We find that without the contrastive loss, the model is unable to converge and performs very badly. Our results are shown in Table 4.3. For all other experiments, we defaulted to using contrastive loss as the supplementary objective.

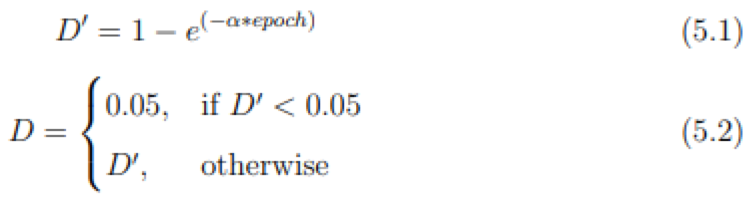
We hypothesize that when used in conjunction, the contrastive loss and triplet ranking loss complement each other. The contrastive loss provides a fine-grained measure of similarity between pairs, aiding in separating positive and negative pairs effectively. The triplet ranking loss then enforces a broader notion of similarity, encouraging the network to understand relative relationships within triplets. This combined approach helps the model to learn a more robust and discriminative feature space, leading to improved retrieval performance.

33. **Examiner's comment:** Page 80: The summary describes that "Converging Tied layers are proposed to increase efficiency and to merge the two disjoint audio and text encoder". However, currently the results do not demonstrate any increase in the efficiency caused by the converging tied layers, since the tested models are (pre-)trained with different data.

    Response: Thank you for this comment. Please refer to related response 31.

34. **Examiner's comment:** Page 84: in formula (5.2), there is D in both left- and right-hand side of the equality. I assume there should be D' in the right-hand side.

    Response: Thank you for this comment. We have fixed the equation.

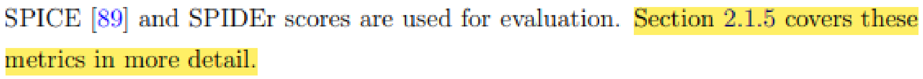


$$D' = 1 - e^{(-\alpha * epoch)} \tag{5.1}$$

$$D = \begin{cases} 0.05, & \text{if } D' < 0.05 \\ D', & \text{otherwise} \end{cases} \tag{5.2}$$

35. **Examiner's comment:** Pages 88-89 describe the procedures for calculating the evaluation metrics. This is redundant with the explanation in Section 2.1.5.

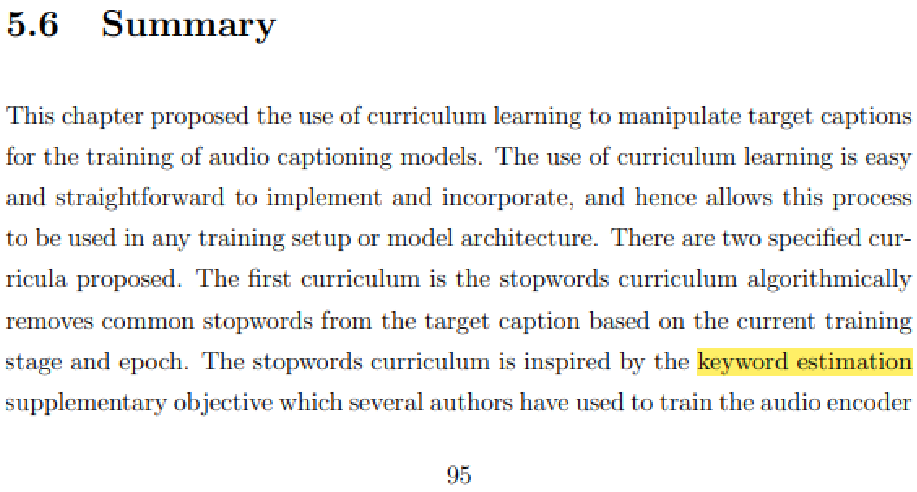
    Response: Thank you for this comment. We have replaced the redundancy with a reference to the section.



SPICE [89] and SPIDEr scores are used for evaluation. Section 2.1.5 covers these metrics in more detail.

36. **Examiner's comment:** Page 94: "Keyword" should be lowercase

    Response: Thank you for this comment. We have fixed the capitalization.



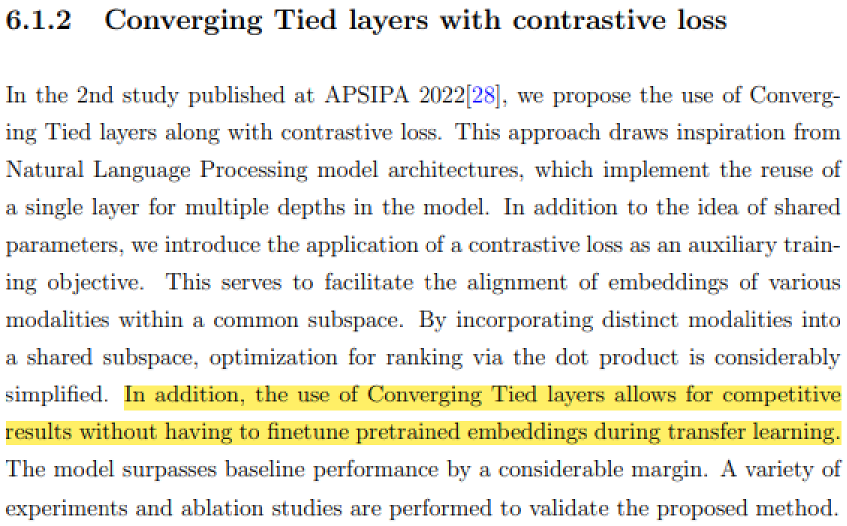
5.6 Summary

This chapter proposed the use of curriculum learning to manipulate target captions for the training of audio captioning models. The use of curriculum learning is easy and straightforward to implement and incorporate, and hence allows this process to be used in any training setup or model architecture. There are two specified curricula proposed. The first curriculum is the stopwords curriculum algorithmically removes common stopwords from the target caption based on the current training stage and epoch. The stopwords curriculum is inspired by the keyword estimation supplementary objective which several authors have used to train the audio encoder

95

37. **Examiner’s comment:** Page 99: The conclusions describe that “Converging Tied Layers significantly improved the performance compared to the baseline”. However, currently the results do not demonstrate any increase in the efficiency caused by the converging tied layers, since the tested models are (pre-)trained with different data. This should be clarified to avoid any false conclusions about the usefulness of the converging tied layers.

    Response: Thank you for this comment. Please refer to response 31 and 33. The efficiency comes from the Converging Tied Layers allowing us to make use of pretrained models without having to further finetune the pretrained model. We have added this statement in.



6.1.2 Converging Tied layers with contrastive loss

In the 2nd study published at APSIPA 2022[28], we propose the use of Converging Tied layers along with contrastive loss. This approach draws inspiration from Natural Language Processing model architectures, which implement the reuse of a single layer for multiple depths in the model. In addition to the idea of shared parameters, we introduce the application of a contrastive loss as an auxiliary training objective. This serves to facilitate the alignment of embeddings of various modalities within a common subspace. By incorporating distinct modalities into a shared subspace, optimization for ranking via the dot product is considerably simplified. In addition, the use of Converging Tied layers allows for competitive results without having to finetune pretrained embeddings during transfer learning. The model surpasses baseline performance by a considerable margin. A variety of experiments and ablation studies are performed to validate the proposed method.

38. **Examiner’s comment:** Page 102: “One” should be lowercase. Bibliography: -The publication information is missing from a large number of publications. For example from the first two pages of the bibliography (109-110), this includes references 5, 16, 17, 18. I am not listing all the publications missing the publication information, but proper publication information (e.g., journal or conference name, issue, year) should be added to all the references -There is a large number arxiv publications. In case of those, one should find out if the arxiv paper is published in a peer-reviewed conference or journal, and instead of arxiv refer to the peer-reviewed version. 7 (7) -Many bibliography items contain capitalization errors (e.g., bert, chatgpt)

    Response: Thank you for this comment. We have gone through each entry and fixed the bibliography errors.

05 November, 2023

_______________________ _________________________

Signature of Student Date