TL;DR Abstractive Summarization with Text-to-Text and Sequence-to-Sequence Transformer Models

Andy Guinto
University of California, Berkeley
aguinto@berkeley.edu

Abstract

Abstractive summarization requires the model to generate words that are both meaningful and coherent. The challenge of generating abstractive summarization from social media based content stems from the large amounts of profanity, acronyms, and slang, demanding a much higher level of contextual understanding. In this study, we will explore how a Text-to-Text and a Sequence-to-Sequence transformer model performs after fine-tuning with the TL;DR dataset as part of the TL;DR challenge. The goal is to explore the feasibility of n-gram based evaluation metrics on social media based summarization and identify key components that allow a model to generate better results. Our fine-tuned models achieved better results when compared to their respective base transformers.

1 Introduction

The rise of digital content has provided an overwhelming amount of information, making it increasingly challenging to consume and comprehend large volumes of text. This surge in information has led to the popularity of short-form media, such as YouTube Shorts, TikTok, Instagram Reels, Facebook Stories, and, notably, the "TLDR" summary. The goal of short-form media is to engage with an audience within a much shorter period of time. However, this begs the question of whether short-form content can effectively convey the same information as their longer counterparts.

"TL;DR," which stands for "Too Long; Didn't Read," is a type of short-form abstractive summary from social media designed to capture the essence of a more comprehensive text, allowing readers to grasp key points without delving into the entire text. Unfortunately, since many studies on abstractive summarization have focused on news or professional environments such as CNN/Daily Mail dataset or SQuAD, this may exhibit a bias toward formal writing and cater predominantly to targeted audiences. In contrast, social media encompasses a broader spectrum of education levels, ages, geolocation, and colloquial and profane language usage which present unique challenges and opportunities for effective summarization.

In this work, which is part of the TL;DR challenge (Voelske et al., 2017) for ACL, the primary objective is to explore and evaluate how well machine summarization performs against social media based content using a combination of n-gram based metrics and qualitative analysis. We will evaluate both a Text-to-Text model such as T5 (Raffel et al., 2020), which attained state of the art performance on the CNN/Daily Mail dataset, and a Sequence-to-Sequence model such as PEGASUS (Zhang et al., 2020), which attained state of the art performance on the XSum dataset.

Fine-tuning both T5 and PEGASUS transformer models on the Reddit TL;DR dataset allows us to understand the strengths and weaknesses of these diverse architectures and understand areas for improvement in handling social media-based content. We will set consistent hyperparameters for both transformer models on both the training and summary generation tasks to maintain a fair comparison. This exploration helps bridge the gap between traditional summarization tasks and the dynamic, informal nature of social media text.

2 Background

Earlier studies (Kim et al., 2010) on N-gram evaluation for summarization found that a modified R-precision outperformed metrics such as BLEU, METEOR, and ROUGE and correlated much better with human scores. However, since this metric relies heavily on counting the n-gram overlap and we don't have the support of generating human scores, we have decided that this metric isn't suitable for our abstractive summarization task.

However, Distinct-N, introduced by Li et al.

Tokenizer	Input	Target
T5-Base	1044	103
PEGASUS-XSUM	945	93

Table 1: The 95th percentile of the generated token counts for each model.

(2016) counts the number of unique n-grams in the generated text as described below. This n-gram metric is useful to measure summarization by measuring diversity for a given *N* sequence of words and might be useful in this study.

Additionally, Viswanath et al. (2021) concluded that the cosine similarities are higher for shorter summaries when trained on T5. Particularly, one of the datasets used in this study was Amazon Food Reviews, which may include informal language similarities when compared to the TL;DR dataset. As a result, we have included cosine similarities in this study to support our n-gram based metrics.

A key inspiration for this work has branched off of a study in 2019 (Gehrmann et al., 2019), which utilized the Reddit TL;DR dataset to evaluate abstract summarization for LSTM and Transformer based models with and without copy attention, as well as pre-trained transformers. This study achieved significantly higher ROUGE scores using transformer based models than LSTM models. Since this study predates T5 and PEGASUS, we will use these scores as part of our baseline.

3 Methods

3.1 Preprocessing

Like most social media platforms, the Reddit TL;DR dataset contained more than three million posts and "tl;dr" (short summaries) containing various counts of profanity, slang, acronyms, and idioms. This is a labeled dataset containing long form posts averaging 270 words and short form (tl;dr) averaging 28 words. Given the skewed distribution of post lengths, we preprocessed the training set to use a maximum length input equal to the 95th percentile of the input token length for each model's base tokenizer as described in Table 1. The training split was a 70, 15, 15 and all training was done on an A100 GPU. With the exception of prepending a *'summarize: '* string for T5, the data was trained as is. We utilized an Adam optimizer with a learning rate of *5e-5* and a cross-entropy loss function.

We first consider the performance of a baseline using the default parameters of T5-Base and PEGASUS-XSUM. We consider the models below for abstractive summarization. All models were trained with 3 epochs, with the exception of the 20 Epochs model. By comparing the same samples for each model type, we can maintain a more accurate comparison. We experimented with a variety of epochs, but we maintained 3 epochs for each model due to time and bandwidth limitations since the larger sample epochs ranged from one day to several days to complete. However, we sampled 10k with 20 epochs for comparison to see any improvements to the performance. We determined the following models would be ideal for capturing how well T5 and PEGASUS perform on this dataset. Adding different numbers of training samples and epochs added diversity when evaluating performance.

Baseline Default T5-Base and PEGASUS-XSum

10k Trained on 10,000 samples and validated on 10,000 other samples.

100k Trained on 100,000 samples and validated on 100,000 other samples.

Full Trained on 2.3 million posts and validated on more than 700,000 other samples.

20 Epochs Models were trained on the same 10,000 samples, but with 20 epochs instead of 3 epochs to see if the model converges better than the other models with more epochs and less samples.

To contextualize how well T5 and PEGASUS learn from this type of content, we kept the summary generation hyperparameters constant across all model types, with the exception of the additional experiments as a result of the generated summaries. For the initial run, every model had `minimum output length` of *0* and a `maximum output length` that exceeds the respective tokenizer maximum. For every model, we generated 500 summaries 3 times and achieved similar results at every attempt.

3.2 N-Gram Metrics

BLEU (Papineni et al., 2002) is one of the most established metrics for evaluating n-grams for summarization tasks. Since BLEU utilizes n-gram precision and incorporated a brevity penalty (1), we expect to see a low BLEU score regardless of

whether we adjust the maximum summary length hyperparameter. The goal is to obtain shorter and engaging summaries, which would be penalized by this metric. [1]

$$BrevityPenalty = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases} \quad (1)$$

METEOR (Banerjee and Lavie, 2005) improved upon the BLEU's limitations by taking the ordering of words into consideration. However, the fragmentation penalty (2) is calculated based on lexical matching of chunks or contiguous sequence of words between the reference and generated summaries. Similar to BLEU, we expect to see a a low METEOR score.[2]

$$\textit{Fragmentation Penalty} = \gamma \left(\frac{chunks}{matchedwords} \right)^{\beta} \quad (2)$$

The **ROUGE** (Lin, 2004) metric is used for evaluating summarization in almost all summarization tasks. Like BLEU and METEOR, it still evaluates how much overlap is captured between the reference and generated summaries, making this metric more suited towards extractive summarization. For our abstractive summarization case, we expect to see a higher ROUGE-1 score than a ROUGE-2. Since we expect the scores to be low, these common n-gram metrics may not be the most ideal metrics for evaluating summaries. A recent study (Barbella et al., 2022) stated that there isn't an effective metric for summarization.

Distinct-N N-Gram Abstractiveness was considered, but was dropped as we saw that it did not provide any insightful information to our study. Instead, we opted to use the Distinct-N (Li et al., 2016) metric as defined in equation (3). This metric focuses on rewarding texts with more unique n-grams, producing more novel combinations of words. As a result, less bias and less redundancy.[3]

$$\textit{Distinct-N} = \frac{\textit{Number of unique N-grams}}{\textit{Total number of N-grams}} \quad (3)$$

[1] r is the length of the reference summary and c is the length of the generated summary.

[2] γ determines the maximum penalty, often set to *0.5*. β determines the relation between fragmentation and the penalty, often set to *3*.

[3] Unfortunately, Distinct-N was not used as a callback function during our training. In a future study with less financial and time constraints, we highly recommend utilizing this metric.

We evaluated our summaries with N-grams 2 and 3, since we felt that it struck a great balance between shorter summaries. Likewise, we felt that using unigrams would highly inflate the results, since the focus is on individual words and not a combination of words. We expect to see a large Distinct-N value across all experiments.

3.3 Cosine Similarity

To support N-Gram Metrics, we want to explore the cosine similarity (4) scores across all models. Contrarily to the N-Gram metrics, the cosine similarity focuses on semantic similarity rather than lexical matching of words, which results in capturing additional context. A previous study (Viswanath et al., 2021) explored using the cosine similarity and found that it performed better for short summaries.[4]

$$\textit{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|} \quad (4)$$

To calculate the cosine similarity of our metrics, we re-tokenize all generated summaries across all models using term frequency inverse document frequency (TF-IDF). Since we are focused on the generated summaries of the models, this provided a computationally efficient way to create a uniform vector representation while neutralizing the differences in the T5 and PEGASUS tokenizers.

4 Results

Table 3 describes the results of the N-Gram metrics for the test set for each model. As expected, the BLEU, ROUGE, and METEOR scores were low due to the references being very abstractive. For these results, we set the max length output to be much higher than the reference summary to not restrict the model when generating the summaries. As expected, training on the entire dataset has mostly outperformed training on a much smaller dataset.

An interesting result is that training on the same 10,000 training samples and 10,000 validation samples have yielded worst N-Gram metric results for 20 epochs than 3 epochs, but better results for Distinct-2 and Distinct-3 as described on Table 2. Looking at the training history, we determined that both T5 and PEGASUS were overfitting when reaching higher epochs. Figure 1 shows expected results, where at each epoch, the loss is decreasing

[4] A and B represent the reference and generated summaries.

and the accuracy is increasing for both training and validation sets, implying that the model is learning.[5] However, Figure 2 shows results where the model may be overfitting at a much earlier epoch.[6]. Consistently, both T5 and PEGASUS models have increasing loss and decreasing accuracy for the validation set, implying that the models are overfitting on the training set.

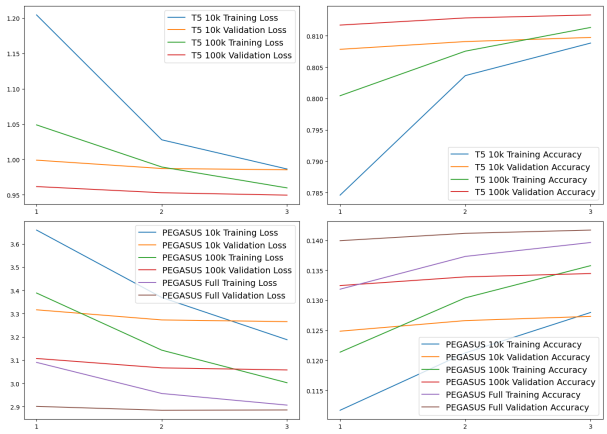

Figure 1: Training History for 3 Epochs.

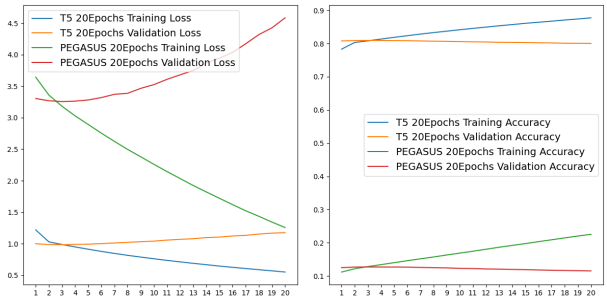

Figure 2: Training History for 20 Epochs.

Additionally, when considering the average length of the summaries T5-20Epochs averages 19.2, PEGASUS-20Epochs at 17.1, T5-Full at 19.1 and PEGASUS-Full at 15.8. This may indicate issues in adding unnecessary verbose output. Qualitative analysis is required to assess this.

Due to time and resource constraints, we generated a cosine similarity heatmap of 10 randomly generated summaries from our test set shown on Figure 4. Across the board, we see inconsistent results. However, there are outliers (i.e. those with high cosine similarity, and low cosine similarity) that we will focus on. The generated summaries for

[5]It is known that the T5-Full is missing on the plots for loss and accuracy. See the Limitations section for more information.

[6]For training with 20 epochs, we disabled the `EarlyStopping` callback

Model	Distinct-2	Distinct-3
T5-Base	94.82	96.67
T5-10k	95.07	96.49
T5-100k	94.33	94.43
T5-Full	92.94	93.62
T5-20Epochs	**97.93***	**98.69***
REF	97.58	97.23
PEGASUS-Base	95.25	95.91
PEGASUS-10k	95.67	93.75
PEGASUS-100k	95.13	93.75
PEGASUS-Full	93.41	92.29
PEGASUS-20Epochs	**96.73**	**96.31**

Table 2: The Distinct-N values of our test set.

both high and low cosine similarities are described on Table 4 and Table 5.

We notice that the summaries with high cosine similarities contain similar sequences of words such as `Muschamp is a good defensive coordinator` and `figure head` which imply high overlap, which explains the higher cosine similarity. Qualitatively, we notice that the summaries generated by T5 and PEGASUS were roughly very similar and generated a more readable summary than the reference summary. For the summaries with the low cosine similarities, we can qualitatively see that each summary was more diverse, meaning that there are less common sequences, but still captured similar meaning. Interestingly, PEGASUS-100k and PEGASUS-Full produced worst qualitative summaries than the PEGASUS-10k (including 20Epochs). We achieved similar results when qualitatively analyzing several other indices in our test set with similar cosine similarities. This is indicative of our overfitting hypothesis or there exists repetitive patterns that did not generalize well for these examples.

Holistically looking at the averages across the generated summaries in the test set, we can see that the T5-Full and PEGASUS-Full outperformed the baseline and the other experimental models as seen on Figure 3.

However, we slightly underperformed when compared to results studied by Gehrmann et al. (2019). We will further experiment our best models (Full) and our worst models (20Epochs). We forcibly generate a shorter summary than the average length of our currently generated summaries across all models. For this, we will set `max tokens`

Model Type	BLEU	R1	R2	RL	METEOR
T5_Base	1.82	16.83	3.52	12.23	**18.69***
T5_10k	2.93	19.07	5.43	15.25	16.98
T5_100k	3.00	19.00	5.51	15.62	16.82
T5_Full	**3.74***	**20.36***	**6.55***	**17.17***	17.85
T5_20Epochs	2.73	18.78	4.62	15.10	16.41
PEGASUS_Base	0.71	11.86	1.74	9.40	9.24
PEGASUS_10k	1.95	17.45	4.94	14.30	14.34
PEGASUS_100K	2.26	18.30	5.51	15.22	15.12
PEGASUS_Full	**2.99**	**19.61**	**6.25**	**16.41**	**16.75**
PEGASUS_20Epochs	1.82	16.77	3.48	13.18	14.17

Table 3: The BLEU, ROUGE, and METEOR results on the TL;DR test set.

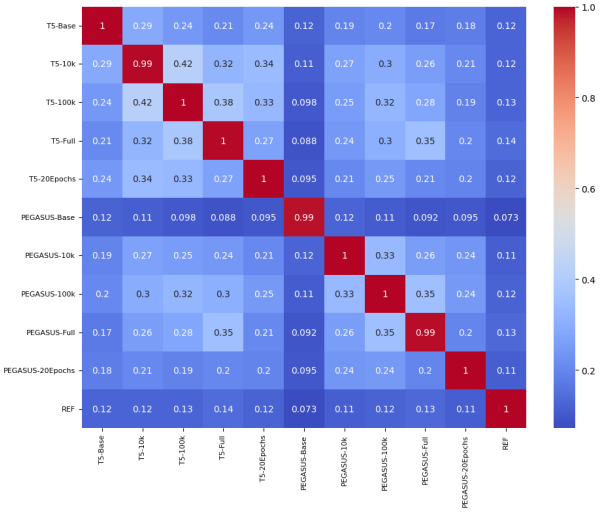

Figure 3: Average cosine similarity across our test set.

to *50* and `maximum output length` to *10*. Table 6 shows the results. The performance was worse than the original generated summaries. Now, looking at the generated summaries of the content used for the high and low cosine similarities on Table 8 and Table 9, we can infer that the models did not capture the entire meaning as described by the content and the reference summary. The qualitative analysis is aligned with the lower n-gram scores.

We will try with a slightly longer summary than the average of the currently generated summaries across all models. For this, we will set `maximum output length to 50` and `minimum output length` to *25*. Table 7 shows the results. These results have outperformed Gehrmann et al. (2019) for ROUGE-1 and METEOR. And now looking at the generated summaries, we can infer that T5 learned to generate a more comprehensive summary than PEGASUS.

While it was expected that the Full models would outperform the other models we experimented with, PEGASUS performed better when fine-tuned to generate shorter summaries. In fact, it generated the same summary when fine-tuned to generate a longer summary. Likewise, T5 performed better when fine-tuned to generate longer summaries.

5 Limitations

During our research, we experienced a hardware failure that corrupted the `tf.keras.callbacks.History` object of our T5-Full model. Despite numerous attempts to recover this file or part of this file through various data recovery techniques, we were unsuccessful. Re-training the model or evaluating the model at each epoch was considered, but the high computational and financial costs exceeded the time and budget constraints of this project. Specifically, training this model alone incurred costs upwards of *$400* and *3-4 days* of training and evaluating. As a result of these constraints, we omitted this information on Figure 1. However, it's important to highlight that the impact of this data loss on our analysis is minimal as it was only used to compare loss and accuracy across epochs. We retained the model weights and have conducted our analysis as planned.

Additionally, we would have liked to conduct more analysis on more summaries than the 500 we studied here. Among the 500 we have generated, we still have almost over 700,000 in our testing set. Also, experimenting with regularization techniques to prevent overfitting on the 20 epochs may have generated more promising results. Both time and financial constraints played a role in the limitations of this study.

6 Conclusion

In this paper, we studied the abstractive summarization performance generated by T5 and PEGASUS when fine-tuned on the TL;DR dataset. Evaluation was done using n-gram based metrics, cosine similarities, and qualitative analysis. For our abstractive summarization task, we trained different models and experimented with different hyperparameters for summary generation and found that T5 performed better on longer summarization, while PEGASUS performed better on shorter summarization.

Future work may involve including other forms of social media based content, experimenting with different architectural layers, and different metrics such as M-TER (Agarwal and Lavie, 2008), as it offers more flexibility in its edit rate and matching, which may possibly perform well against the reference summaries (tl;dr) generated by humans. We hope this analysis will encourage the further development of metrics to evaluate abstractive summarization, specifically for evaluating social media based summaries.

References

Abhaya Agarwal and Alon Lavie. 2008. Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with human rankings of machine translation output. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 115–118, Columbus, Ohio. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, MI, USA. Association for Computational Linguistics.

Marcello Barbella, Michele Risi, Genoveffa Tortora, and Alessia Auriemma Citarella. 2022. Different metrics results in text summarization approaches. In *Proceedings of the 11th International Conference on Data Science, Technology and Applications (DATA 2022)*, pages 31–39. SCITEPRESS.

Sebastian Gehrmann, Zachary M. Ziegler, and Alexander M. Rush. 2019. Generating abstractive summaries with finetuned language models. In *Proceedings of The 12th International Conference on Natural Language Generation*, pages 516–522, Tokyo, Japan. Association for Computational Linguistics.

Su Nam Kim, Timothy Baldwin, and Min-Yen Kan. 2010. Evaluating n-gram based evaluation metrics for automatic keyphrase extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 572–580, Beijing, China.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of NAACL-HLT*, pages 110–119. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL Workshop on Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero Dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA, USA. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Karthik Viswanath, L Naveen, Ananda Raj, S Rajalakshmi, and Angel Deborah. 2021. Abstractive text summarizer: A comparative study on dot product attention and cosine similarity. In *Proceedings of the Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 20–21, Chennai, Tamil Nadu, India. IEEE.

Michael Voelske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl;dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11328–11339.

A Appendix

The following appendix contain referenced figures from this study.

This is a section in the appendix.

A.1 Cosine Similarity - 10 Random

Due to the size of the figures and tables, we kept them on separate pages.

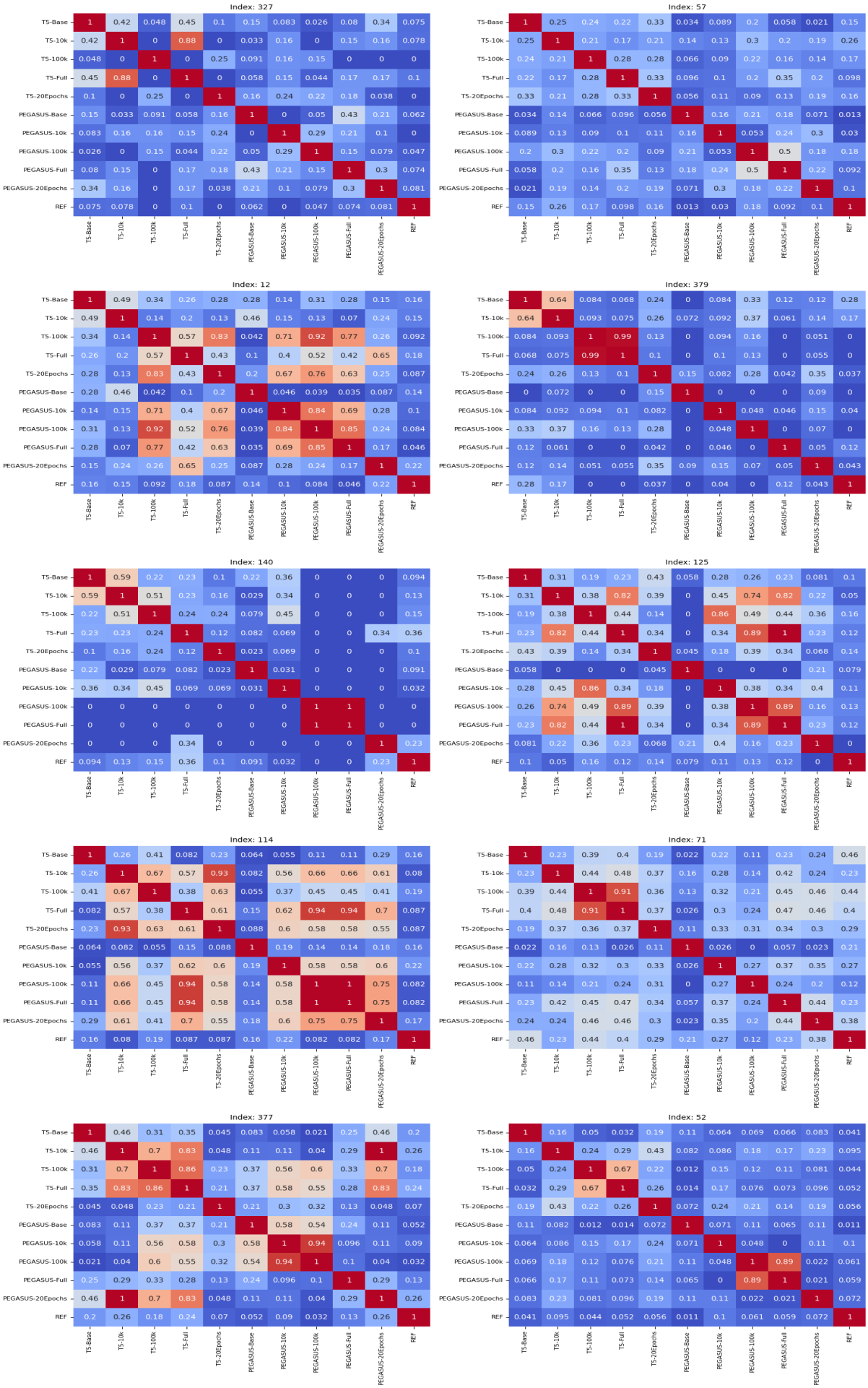

Figure 4: The cosine similarity heatmap for 10 randomly generated summaries in the test set and the reference.

A.2 High Cosine Similarity

Content: I've said this before, and I'll say it again: Muschamp is a good defensive coordinator. Is he a good head coach? If he got a good offensive coordinator that could compete with modern college football, yes, but he would be acting as a figure head. As much as I hate to say that, it's true. 'Champ has 0 offensive knowledge, and can only work a defense.

Model	**Generated Summary**
T5-10k	Muschamp is a good defensive coordinator, but he's a figure head.
T5-100k	Muschamp is a good defensive coordinator, but he would be acting as a figure head.
T5-Full	Muschamp is a good defensive coordinator, not a good head coach.
T5-20Epochs	Muschamp is a good defensive coordinator. He's a figure head.
PEGASUS-Base	Will Muschamp has been named the new head coach at Florida.
PEGASUS-10k	Will Muschamp is a good defensive coordinator.
PEGASUS-100k	Muschamp is a good defensive coordinator, but not a good head coach.
PEGASUS-Full	Muschamp is a good defensive coordinator, but not a good head coach.
PEGASUS-20Epochs	Will Muschamp is a good head coach, but he is not a good offensive coordinator.
REF	If Roper pans out, people will look at Will as a good coach. If Roper doesn't pan out, 'Champ will be fired. All as a result of being a figure head type HC.

Table 4: High cosine similarity summaries. Heatmap is *Index 114* in Figure 4 in the appendix.

A.3 Low Cosine Similarity

Content: Spend 20-30 minutes warming up: scales, Hanon exercises, maybe even the Czerny etudes. While you play, make sure to focus on the music, don't let your mind wander too much. Music is a great way to escape reality, so don't focus on your life, just think about the musical themes in your piece. Enjoy! Loving what you play and having a great attitude towards it can make the difference between winning and losing a competition– no matter how perfect your technique is. Make sure you have an image in mind for each part of your pieces, so you can properly distinguish the sound of passages that, for example would sound like a torrent, from that of a flowing breeze.

Model	**Generated Summary**
T5-Base	music is a great way to escape reality, so don't let your mind wander too much. don't focus on your life, just think about the musical themes in your piece.
T5-10k	Don't let your mind wander too much, focus on the music. Enjoy!
T5-100k	Enjoy the music, focus on the music.
T5-Full	Warm up, enjoy the music, have an image in mind for each part of your piece.
T5-20Epochs	Enjoy your music, it can make all the difference between winning or losing.
PEGASUS-Base	This is a great way to get into the swing of things.
PEGASUS-10k	Don't worry about technique, just focus on the music.
PEGASUS-100k	Practice, practice, practice.
PEGASUS-Full	practice, practice, practice.
PEGASUS-20Epochs	Warm up, have fun!
REF	Warm up. Enjoy! Create images for the different parts of your pieces.

Table 5: Low cosine similarity summaries. Heatmap is *Index 140* in Figure 4 in the appendix.

A.4 Updated Metrics Table

Model Type	BLEU	R1	R2	RL	METEOR
T5_Full	**0.17**	**14.60**	**4.58**	**13.12**	**8.56**
T5_20Epochs	0.15	12.70	3.02	11.35	7.18
PEGASUS_Full	**0.24**∗	**15.52**∗	**4.92**∗	**13.98**∗	**9.53**∗
PEGASUS_20Epochs	0.15	12.35	3.04	11.08	7.14

Table 6: The BLEU, ROUGE, and METEOR results on the TL;DR test set using a shorter summary parameters.

Model Type	BLEU	R1	R2	RL	METEOR
T5_Full	**4.24**∗	**20.25**∗	**6.16**∗	**16.54**∗	**19.68**∗
T5_20Epochs	3.13	19.0	4.34	14.58	17.94
PEGASUS_Full	**3.67**	**19.78**	**5.8**	**15.54**	**19.54**
PEGASUS_20Epochs	2.44	17.57	3.58	13.3	13.3

Table 7: The BLEU, ROUGE, and METEOR results on the TL;DR test set using slightly longer summary parameters.

Model	Generated Summary
T5-Full	Muschamp is a good defensive coordinator
T5-20Epochs	Muschamp is a good defensive coordinator
PEGASUS-Full	Muschamp is a good defensive coordinator,
PEGASUS-20Epochs	Will Muschamp is a good head coach
REF	If Roper pans out, people will look at Will as a good coach. If Roper doesn't pan out, 'Champ will be fired. All as a result of being a figure head type HC.

Table 8: High cosine similarity summaries from heatmap is *Index 114* in Figure 4, but adjusted for shorter summaries.

Model	Generated Summary
T5-Full	Warm up, enjoy the music, have an
T5-20Epochs	Enjoy your music, focus on the music,
PEGASUS-Full	practice, practice, practice.
PEGASUS-20Epochs	Warm up, have fun!
REF	Warm up. Enjoy! Create images for the different parts of your pieces.

Table 9: Low cosine similarity summaries from heatmap is *Index 140* in Figure 4, but adjusted for shorter summaries.

Model	Generated Summary
T5-Full	Muschamp is a good defensive coordinator, but he's not a good head coach. 'Champ is a good offensive coordinator, but not a good head coach.
T5-20Epochs	Muschamp is a good defensive coordinator, but he's a figure head for modern college football.
PEGASUS-Full	Muschamp is a good defensive coordinator,
PEGASUS-20Epochs	Will Muschamp is a good head coach
REF	If Roper pans out, people will look at Will as a good coach. If Roper doesn't pan out, 'Champ will be fired. All as a result of being a figure head type HC.

Table 10: High cosine similarity summaries from heatmap is *Index 114* in Figure 4, but adjusted for slightly longer summaries.

Model	Generated Summary
T5-Full	Warm up, enjoy the music, have an image in mind for each part of your piece, don't let your mind wander.
T5-20Epochs	Enjoy your music, it can make all the difference between winning or losing. Don't be afraid of music too much.
PEGASUS-Full	practice, practice, practice.
PEGASUS-20Epochs	Warm up, have fun!
REF	Warm up. Enjoy! Create images for the different parts of your pieces.

Table 11: Low cosine similarity summaries from heatmap is *Index 140* in Figure 4, but adjusted for slightly longer summaries.