Automated Lecture Video Summarization

Michael Kreager
School of Electrical and Computer Science
Faculty of Engineering
University of Ottawa
mkrea053@uottawa.ca

Abstract

We present a sophisticated, yet pragmatic, solution for automatic summarization of lecture videos combining the state-of-the-art long document Transformer model, LED, with the long-proven graph-based model, TextRank. The model outputs a web page that includes a high-level abstract summary generated by LED, with key words/phrases and a detailed summary body extracted by TextRank. Summary body sentences are hyperlinked to the source sentences in the full transcript, which in turn are hyperlinked to the timestamps in the source video.[1] Our statistical (ROUGE) and qualitative assessment shows that our model provides a significant improvement over BERT Extractive Summarizer.

1 Introduction

The onset of the Covid-19 global pandemic has accelerated the shift towards online learning (Dhawan, 2020). In this paper, we propose a solution for automatic summarization of lecture videos to aid students in e-learning. We demonstrate the solution on a university-level lecture posted to YouTube, with a running time of 1 hour 32 minutes (DeepMind, 2020). The lecture transcript is a long document with a total word count of 14,893.

1.1 Summarization

Automatic summarization is an important and challenging task in natural language processing (NLP) (Huang et al., 2020). Automatic summarization is generally categorized as either extractive or abstractive (Carenini and Cheung, 2008). The goal of extractive summarization is to reduce the source text while retaining meaningful content by direct selection of the most important sentences. Abstractive summarization has a similar objective but allows for paraphrasing and the introduction of words not necessarily found in the source text to produce outputs of better fluency and coherency (than extractive summaries).

1.2 Transformers

The Transformer model architecture, introduced in 2017, demonstrated that state-of-the-art machine translation performance could be achieved using self-attention mechanisms without the added computational overhead of recurrence or convolutions used by prior high-performing models, such as long short-term memory, recurrent neural networks, gated recurrent units and convolutional neural networks (Vaswani et al., 2017). Many variations of the Transformer model have since emerged showing impressive performance on various other NLP tasks, including summarization, text generation and question answering (e.g. Devlin et al., 2019; Lewis et al., 2019; Brown et al., 2020; Zhang et al., 2020a).

Transformers, though very technically impressive, are not without problems. Aside from significant environmental and ethical concerns (Bender et al., 2021), Transformers have shown a tendency to generate text that is factually inconsistent and unfaithful to the source text (in the case of summarization) (Maynez et al., 2020). An example of unfaithful and inaccurate output is detailed in Appendix A.

A few Transformer models with relevance to this study are briefly discussed in the following paragraphs.

BERT is an early implementation of Transformer architecture introducing deep bi-directional encoding to learn language context looking forward as well as backward (Devlin et al., 2019).

BART is a denoising autoencoder Transformer model that builds on BERT's bidirectional encoder and combines it with an autoregressive decoder

[1] The code for this paper is available here: https://github.com/mkreager/nlp-summarization

(Lewis et al., 2019). BART is particularly successful with abstractive summarization tasks.

PEGASUS is a Transformer encoder-decoder model with a specific focus on pre-training objectives for summarization (Zhang et al., 2020a). Whereas BERT and BART mask individual tokens or short spans, PEGASUS masks entire sentences for prediction.

Longformer is a Transformer model for processing long documents that uses an efficient local plus global sparse attention mechanism with sliding window (Beltagy et al., 2020).

LED or Longformer-Encoder-Decoder, is a variation of the Longformer model that uses BART's parameters and architecture (Beltagy et al., 2020). LED is specifically designed for long document sequence-to-sequence tasks, including summarization.

1.3 TextRank

TextRank is an unsupervised graph-based ranking model for text processing that was introduced in 2004 (Mihalcea and Tarau, 2004). TextRank is computationally efficient and, unlike Transformers, does not require resource-intensive pre-training.

PositionRank is another unsupervised graph-based text ranking model (Florescu and Caragea, 2017). Unlike TextRank, PositionRank incorporates the relative positioning of words into the ranking.

Biased TextRank is a variation of TextRank that introduces an input focus to the ranking (Kazemi et al., 2020). Refer to Appendix B for an example and analysis of Biased TextRank output.

2 Related Work

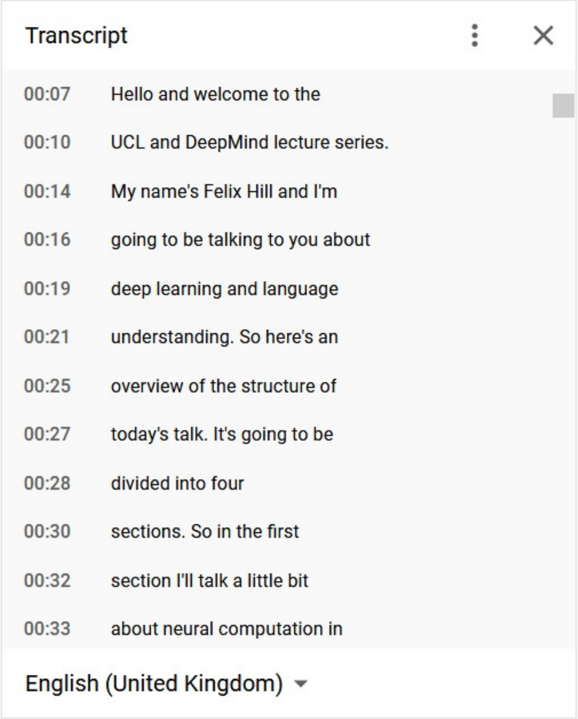
The NLP leaderboard rankings present many alternative models for extractive and abstractive summarization (Ruder). In addition to Longformer/LED, a few other Transformers for long document processing have recently emerged, including Reformer (Kitaev et al., 2020) and BigBird (Zaheer et al., 2021).

BERT Extractive Summarizer is an extractive summarization model that utilizes BERT for text embeddings and K-Means clustering for summary sentence selection (Miller, 2019). BERT Extractive Summarizer was introduced specifically for the summarization of long lecture videos. In Section 4, we compare our model to the BERT Extractive Summarizer model.

3 Proposed Solution

The video transcript from Lecture 7 of the *DeepMind x UCL Deep Learning Lecture Series*, is used as the input to our model (DeepMind, 2020). The beginning section of the transcript from YouTube is shown in Figure 1.

For our proposed solution to be useful, we acknowledge that students will require the output to reflect accurate information. Hence, we prioritize factual and faithful accuracy over fluency and coherence. We also include some techniques to compensate for lacking fluency and coherence.



Transcript

00:07 Hello and welcome to the
00:10 UCL and DeepMind lecture series.
00:14 My name's Felix Hill and I'm
00:16 going to be talking to you about
00:19 deep learning and language
00:21 understanding. So here's an
00:25 overview of the structure of
00:27 today's talk. It's going to be
00:28 divided into four
00:30 sections. So in the first
00:32 section I'll talk a little bit
00:33 about neural computation in

English (United Kingdom)

Figure 1: Beginning section of the lecture video transcript on YouTube (DeepMind, 2020).

3.1 Preprocessing

We begin by pulling the raw transcript data into a dataframe within our interactive Python (IPython) notebook. We then prepend timestamps to phrases in a new column in the dataframe. Each of the original text and time-prepended text columns are joined into continuous strings.

The strings are then segmented into sentences using the Natural Language Toolkit (NLTK) (Bird et al., 2019) and inserted into a new dataframe. We then have segmented original sentences matched up with segmented timestamp-embedded sentences. An example of the first sentence of the transcript with embedded timestamps is illustrated in Figure 2. We can then extract the first timestamp in each sentence for future use.

Note that the NLTK sentence segmenter provides a predictable output that allows us to easily match original sentences with corresponding timestamp-embedded sentences. The spaCy DependencyParser (Honnibal et al., 2020), on the other hand, recognizes that the embedded timestamps do not belong in the sentences and attempts to split them out (especially where timestamps are embedded before the starts or after the ends of sentences). We therefore override the spaCy DependencyParser to provide pre-segmented sentences from NLTK to our processing pipeline.

> t=13 My name's Felix Hill and I'm t=15 going to be talking to you about t=18 deep learning and language t=20 understanding.

Figure 2: Example of a segmented sentence with embedded timestamps.

3.2 Extractive Summarization

We use the PyTextRank (Nathan, 2016) implementation of TextRank with our spaCy pipeline to extract the top ranked words/phrases and sentences from the pre-segmented input sentences. We set the limit on returned words/phrases to 20 and the limit on sentences to 30% of the original sentence count. We use a ratio of 5:1 for ranked phrases to sentences for the graph model vertices. The resulting output is 144 sentences, returned in order of original appearance (not ranked order), containing 6434 words (57% fewer than the original text).

3.3 Abstractive Summarization

We take the TextRank output sentences from the previous step as input to a pre-trained LED model, fine-tuned on the PubMed dataset – model: patrickvonplaten/led-large-16384-pubmed from the HuggingFace Transformers library (Wolf et al., 2020). PubMed is a dataset of scientific papers that is well-suited to long-range summarization training objectives (Cohan et al., 2018). A short abstractive summary of the transcript is output from this step.

3.4 Document Output

The final output is a minimally-formatted HTML web page with the following sections:

1. Overview – a short introduction;
2. Abstract – the abstractive summary;
3. Keywords/phrases – top 20 words/phrases;
4. Summary – the extractive summary;
5. Full Transcript – the full video transcript.

Sentences are grouped into paragraphs in the 'Summary' section based on their positional locations. Paragraphs provide visual cues to the user for where context gaps may exist. Long paragraphs indicate several sentences in close proximity with minimal pruning between them. Short paragraphs and 'orphaned' sentences suggest that more context may be needed.

Sentences in the 'Summary' section are hyperlinked to the 'Full Transcript' section. Sentences in the 'Full Transcript' section are hyperlinked to the video at the approximate time of utterance, using the timestamps that we extracted in the preprocessing step. The hyperlinks are provided as a convenience for the user who wishes to obtain additional context or clarification.

4 Evaluation

To evaluate our model, we conduct some qualitative analysis of the abstractive and extractive summaries, as well as a statistical evaluation of the extractive summary.

4.1 Human Summary

To evaluate the performance of our extractive summarization technique, we first must create a human summary of the most important sentences, targeting the same sentence count as the TextRank output. The human summary is produced in three passes through the transcript. In the first pass, mostly short and uninformative sentences are pruned. In the second pass, we remove sentences with information that is substantially covered elsewhere. In the final pass, some important information inevitably must be dropped.

4.2 Extractive Summary Evaluation

Miller (2019) generally found the BERT Extractive Summarizer model to be superior to TextRank for producing coherent summaries of lecture videos. To compare with our model, we follow similar steps to input the data to the BERT Extractive Summarizer model and limit the number of returned sentences to be equivalent to our TextRank output.

Considering the first few sentences of the BERT Extractive Summarizer output, as shown in Figure 3, we note that while the coherence is good it is arguably not any better than the TextRank output. To compensate for coherence and context gaps in our model, we organize the output into 'paragraphs' as outlined in subsection 3.4 to provide visual cues to the user when additional context may be needed.

Additionally, the BERT Extractive Summarizer output includes sentences that were eliminated by both TextRank and human summarization. This would indicate that the BERT Extractive Summarizer is favouring certain less important sentences, and sacrificing other more important sentences, in its selection.

> **Hello and welcome to the UCL and DeepMind lecture series.** My name's Felix Hill and I'm going to be talking to you about deep learning and language understanding. **So here's an overview of the structure of today's talk. It's going to be divided into four sections.** And that model is the transformer which was released in 2018 and then in section three I'll go a bit deeper into a particular application of the transformer, that's the well known BERT model, and BERT in particular is an impressive demonstration of unsupervised learning and the ability of neural language models to transfer knowledge from one training environment to another.

Figure 3: First few output sentences from BERT Extractive Summarizer. Sentences in **bold** are not found in the human or TextRank summaries.

ROUGE (Lin, 2004) is a statistical evaluation metric that is commonly used to evaluate the quality of automatic summarization outputs. We use ROUGE-1, ROUGE-2 and ROUGE-L to measure the overlap of unigrams, bigrams and longest common subsequences, respectively, between the automatic outputs and the human summary. As indicated in Table 1, TextRank slightly outperforms PositionRank and significantly outperforms BERT Extractive Summarizer on all three measures.

Model	**R1**	**R2**	**RL**
BERT Ext. Summ.	65.21	44.06	37.71
PositionRank	85.64	69.14	62.77
TextRank	**86.85**	**69.96**	**62.88**

Table 1: ROUGE score comparisons for BERT Extractive Summarizer, PositionRank and TextRank models.

We compare the execution time of the models using the Python `%%timeit` magic function. Results, as indicated in Table 2, show that TextRank is orders of magnitude faster than BERT Extractive Summarizer. TextRank executes quickly on a CPU, whereas a CPU is impractical for BERT Extractive Summarizer execution.

4.3 Abstractive Summary Evaluation

A qualitative assessment of the abstractive output of the LED model indicates good results overall. The model largely produces a factual, faithful and fluent summary. In some instances, the output sentences have improved coherence over the originals. The model produces a good high level summary of the lecture, with much of the content closely resembling sentences from the introductory section of the lecture. Detailed observations are presented in Table 3.

5 Limitations and Opportunities

As this study focused on a single video transcript, further testing is needed to evaluate how the model will generalize to other lecture videos. Additionally, the transcript used in the study was manually edited. It is likely that further work will be required in order to handle machine-generated transcripts with limited or no punctuation and other mistakes.

The model *may* be improved by including techniques for coreference resolution for substitution of entity names as demonstrated in the example in Figure 4. This could help to reduce the reliance on context provided in prior sentences. However, this could come a cost of reduced fluency and coherence.

Another potential improvement could be to apply sentence simplification, splitting and rephrasing as shown in Figure 5. In this example, one very long sentence could be split into four shorter sentences with minimal rephrasing. This could result in shortened extractive summaries while maintaining an equivalent level of usefulness. Some potential strategies have been proposed by Martin

Model	Processor	Time	Runs
BERT Extractive Summarizer	GPU	27.7 s	1 loop, best of 5
BERT Extractive Summarizer	CPU	5 min 1s	1 loop, best of 5
TextRank	**CPU**	**165 ms**	**10 loops, best of 5**

Table 2: Execution time comparisons for BERT Extractive Summarizer and TextRank models.

LED Output Sentence	Observations
Deep learning and language understanding is an enormous area of research in machine learning and neural computation.	Coherent and fluent, but not particularly faithful. Mostly factual.
It has been shown that deep learning has been able to improve performance on a lot of language processing applications over the last few years, so it raises the question of why deep learning, and models which have this neural computation at the heart of their processing, have been so effective in language processing.	Faithful and fluent. Close resemblance to original sentence, with improvement to coherence.
In the first section of this lecture we give an overview of neural computation in general and language in general, and then we give some idea of why neural computation, deep learning or language might be an appropriate fit to come together and produce the sort of improvements and impressive language processing performance that we have seen over the past few years.	Mostly faithful and fluent. Close resemblance to original sentence. Minor improvement to coherence. Minor alteration to meaning by replacing 'and' with 'or'.
In particular, we focus in on one particular neural language model, which we think is quite representative of many of the principles that govern all neural language models.	Faithful, fluent and coherent. Close resemblance to original sentence. 'Particular' is repeated.
And that model is the transformer which was released in 2018 and then in section three we go a bit deeper into a particular application of the transformer, that's the well known BERT model, and BERT in particular is an impressive demonstration of unsupervised learning and the ability of neural language language models to transfer knowledge from one training environment to another.	Faithful, fluent and coherent. Nearly a direct copy of the original sentence.
And in the final section, we take a bit more of a look towards the future of language understanding and deep learning.	Faithful, fluent and coherent.
To do that we delve into some work that's been done at DeepMind on grounded language learning, where we study the acquisition of language in deep neural networks that have the ability to interact and move around simulated environments.	Faithful, fluent and coherent. This and the previous sentence were split from a single long original sentence.

Table 3: Abstractive sentence output from LED with qualitative observations.

et al. (2020), Narayan et al. (2017), Aharoni and Goldberg (2018) and Zhang et al. (2020b).

Further investigation is required for target limits of extractive summary output length and quantity of ranked phrases. We have somewhat arbitrarily chosen to limit the number of sentences to 30% of the original transcript while limiting ranked phrases to 5x the sentence limit. An open question is whether we could automatically determine optimal limits.

Finally, we could consider including automatic evaluation for abstractive summarization output, such as BLEURT (Sellam et al., 2020), BERTScore (Zhang et al., 2020c), or MoverScore (Zhao et al., 2019), to supplement or replace human qualitative

Original Sentence:
The model might require to separate them out into the different parallel heads if words have various different senses.

Coreference Resolved:
The BERT model might require to separate words out into the different parallel heads if words have various different senses.

Figure 4: Example of coreference resolution.

Original Sentence:
So these are kind of wacky effects of how meanings interact when two words come together and it's not necessarily easy to explain them in a model which treated every pair of words fed into that model with exactly the same function to combine their meanings, it very much seems to me that what's instead happening is that whatever function is combining the meanings is taking into account the individual meanings of the components going into that function and in, in additional, additionally, that function may well need to take into account a wider knowledge of typical things we might encounter in the world and how their properties might fit together under the constraints of the world as we know it.

Simplified and split:
These are kind of wacky effects of how meanings interact when two words come together. It's not necessarily easy to explain them in a model which treated every pair of words fed into that model with exactly the same function to combine their meanings. It seems that what's instead happening is that whatever function is combining the meanings is taking into account the individual meanings of the components going into that function. Additionally, that function may well need to take into account a wider knowledge of typical things we might encounter in the world and how their properties might fit together under the constraints of the world as we know it.

Figure 5: Example of text simplification, splitting and rephrasing.

assessment.

6 Conclusion

We propose a lecture video summarization model that combines state-of-the-art long-range abstractive summarization techniques with older, yet effective, extractive summarization methods. Our proposed solution uses TextRank in combination with LED (Longformer-Encoder-Decoder) to output an easily navigable web page that links summary sentences to the source video timestamps. We conduct analyses to show that our model outperforms the BERT Extractive Summarizer model.

Acknowledgements

I would like to thank Professor Hussein Al Osman for offering to supervise this project and for providing valuable support, feedback and suggestions along the way.

References

Roee Aharoni and Yoav Goldberg. 2018. Split and Rephrase: Better Evaluation and Stronger Baselines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 719–724, Melbourne, Australia. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 610–623, New York, NY, USA. Association for Computing Machinery.

Steven Bird, Ewan Klein, and Edward Loper. 2019. *Natural Language Processing with Python*. n.p.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Giuseppe Carenini and Jackie C. K. Cheung. 2008. Extractive vs. NLG-based Abstractive Summarization of Evaluative Text: The Effect of Corpus Controversiality. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 33–41, Salt Fork, Ohio, USA. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

DeepMind. 2020. DeepMind x UCL | Deep Learning Lectures | 7/12 | Deep Learning for Natural Language Processing.

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

Shivangi Dhawan. 2020. Online Learning: A Panacea in the Time of COVID-19 Crisis. *Journal of Educational Technology Systems*, 49(1):5–22. Publisher: SAGE Publications Inc.

Corina Florescu and Cornelia Caragea. 2017. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What Have We Achieved on Text Summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online. Association for Computational Linguistics.

Ashkan Kazemi, Verónica Pérez-Rosas, and Rada Mihalcea. 2020. Biased TextRank: Unsupervised Graph-Based Content Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1642–1652, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. *arXiv:2001.04451 [cs, stat]*. ArXiv: 2001.04451.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv:1910.13461 [cs, stat]*. ArXiv: 1910.13461.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable Sentence Simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

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

Derek Miller. 2019. Leveraging BERT for Extractive Text Summarization on Lectures. *arXiv:1906.04165 [cs, eess, stat]*. ArXiv: 1906.04165.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and Rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616, Copenhagen, Denmark. Association for Computational Linguistics.

Paco Nathan. 2016. PyTextRank, a Python implementation of TextRank for phrase extraction and summarization of text documents.

Sebastian Ruder. Summarization.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International*

Conference on Neural Information Processing Systems, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big Bird: Transformers for Longer Sequences. *arXiv:2007.14062 [cs, stat]*. ArXiv: 2007.14062.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv:1912.08777 [cs]*. ArXiv: 1912.08777.

Li Zhang, Huaiyu Zhu, Siddhartha Brahma, and Yunyao Li. 2020b. Small but Mighty: New Benchmarks for Split and Rephrase. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1198–1205, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020c. BERTScore: Evaluating Text Generation with BERT. *arXiv:1904.09675 [cs]*. ArXiv: 1904.09675.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. *arXiv:1909.02622 [cs]*. ArXiv: 1909.02622.

A Unfaithful and Inaccurate Output

The summary output in this section was generated by the google/pegasus-multi_news model from the HuggingFace Transformers library (Wolf et al., 2020). The input to the model was the 'introductory' section of the transcript (459 words).

Several factual inaccuracies and unfaithful passages were output as signified by the coloured text. Notably, Felix Hill is a Research Scientist at DeepMind (as of the time of the video publication) and not a professor of computer science and engineering at University College London. Nor is Felix Hill the co-founder and CEO of DeepMind.

> Felix Hill is a professor of computer science and engineering at University College London and the co-founder and CEO of DeepMind, the artificial-intelligence research and development firm that's been at the forefront of advances in the field of deep learning. In a recent lecture at UCL, Hill gave a presentation on how his team has developed a language-learning system that

Figure 6: An example of unfaithful and factually-inaccurate summarization from a pre-trained PEGASUS model.

UCL is an acronym for University College London. 'UCL' is referenced in the transcript, but nowhere in the text is 'University College London' found. The model has likely learned this connection during training. Also, it may be fair to describe DeepMind as 'the artificial-intelligence research and development firm that's been at the forefront of advances in the field of deep learning'; however, this is not indicated in the source text. And finally, the last sentence in the output is entirely hallucinated by the model.

B Biased TextRank Example

The word 'reinforcement' appears exactly three times in the video transcript. Figure 7 shows the output of Biased TextRank when a bias is placed on the word 'reinforcement' and the number of returned sentences is limited to three. Figure 8 shows the one sentence containing the word 'reinforcement' that is not returned by the model.

Interestingly, the top-ranked sentence does not contain the word 'reinforcement'. However, the phrase 'deep neural networks that have the ability to interact and move around simulated environments' provides somewhat of a description of reinforcement learning. A contributing factor may be that the word 'language' appears frequently in the top twenty key phrases from TextRank. Another possible explanation may be that 'reinforcement' and 'learning' are collocated in the transcript for every occurrence of 'reinforcement' and the word 'learning' appears multiple times in the top-ranked sentence.

And then in the final section, we'll take a bit more of a look towards the future of language understanding and deep learning and to do that we'll delve into some work that's been done at DeepMind on grounded language learning, where we study the acquisition of language in deep neural networks that have the ability to interact and move around simulated environments.

And, of course, in the world of learning when it comes to jointly learning language and behaviour, which involves often reinforcement learning on those sorts of tasks are techniques for having agents develop a more robust understanding of their surroundings and possibly import what's known as a model of their world.

So this is a different type of problem most typically faced by agents which are trained with reinforcement learning.

Figure 7: Sentences **returned** by Biased TextRank.

And importantly, being able to answer these questions requires a particular type of knowledge, that's propositional knowledge, the knowledge, the ability to tell whether something's true or false in our environment and that's often contrasted especially by philosophers with procedural knowledge, which is just the sort of instinctive knowledge that maybe a reinforcement learning agent would naturally have when it learns to solve control problems in a very fast and precise way.

Figure 8: Sentence **not returned** by Biased TextRank.