ON-DEVICE VIDEO SUMMARIZATION

Deepak Gouda, Kriti Arora, Mohit Aggarwal, Vishnu Jaganathan
Georgia Institute of Technology
{deepakgouda,kriti,mohit7,vjaganathan3}@gatech.edu

ABSTRACT

Video summarization is a topic of interest, due it its ability to make users more efficient at digesting information without watching entire videos. Key points from the video can be extracted, and answers for specific questions can be extracted based on the information contained. Previous approaches have tried recurrent architectures or other large-scale models for video understanding, but compute resources for these models often require datacenter GPUs or cloud based models to run. In this paper, we demonstrate that video understanding can be performed on-device. We introduce a novel pipeline to perform video summarization and Q&A tasks on smartphones. We focus on summarization of lecture videos, first creating context by extracting info from slides with OCR on certain keyframes and turning the audio track into text with Vosk. We use BERT-QA to provide extractive answers for questions and fine tune GPT-2 on the CNN-Dailymail dataset to enable GPT to produce summaries of our created context. Overall, this pipeline enables fast summarization and Q&A of lecture videos on device.

1 INTRODUCTION

Our objective is to create a system for on-device video analysis that prioritizes data privacy and reduces dependency on cloud processing, all the while producing concise video summaries and attending to user requests. Our technology enables viewers to rapidly grasp the essentials of videos without having to watch them through to the conclusion by decreasing network data consumption. Our framework also allows the user to ask subsequent questions and get responses based on the video content. This maximizes output while satisfying the present need for speedy, secure information access. By carrying out complex video analysis procedures locally and offering a seamless, rapid experience, we put user privacy first. This user-centric design redefines how people watch videos and paves the way for on-device processing to take center stage as the new norm. The primary goal is to enhance user interaction with video content by offering a dynamic and responsive experience. Our system's architecture has been thoughtfully designed to handle challenging video processing tasks locally on the device, ensuring a seamless and rapid user experience. This strategy eliminates any potential hazards associated with cloud-based solutions and retains control over sensitive data, all while considerably protecting user privacy and enhancing productivity.

2 PROBLEM STATEMENT

2.1 PREVIOUS WORKS

Video understanding is a significant area in the field of multimedia and computer vision. Its objective is to condense lengthy video content into shorter forms or text descriptions without losing essential information. This process aids in efficient video browsing, retrieval, and indexing, and has seen various approaches and developments over the years.

Current approaches often use recurrent models such as RNNs or LSTMs Rumelhart & McClelland (1987); Hochreiter & Schmidhuber (1997). Video Summarization with Long Short-term Memory Zhang et al. (2016) uses the LSTM to select keyframes and subshots of a video. The LSTM is trained on a dataset where videos that are manually annotated for their keyframes. Hierarchical Recurrent Neural Network for Video Summarization Zhao et al. (2017) uses a heirarchical RNN to process the frames of longer videos without a proportionally increasing GPU footprint. More recent

approaches utilize transformers, such as Multimodal Frame-Scoring Transformer for Video Summarization Park et al. (2023) which utilizes a transformer architecture to process the multimodal data from a video and improves performance on keyframe extraction and summary tasks.
The issue with all of these recurrent and transformer based approaches is their need of human annotated data and relatively large GPU memory footprint. With the emergence of Large Language Models (LLMs) and chatbots, developers have started to build pipelines for processing audio and visual information from short videos, but these solutions still rely heavily on putting together several cloud APIs, which are often pay-to-access, often have significant latency, and make no guarantees about keeping uploaded data confidential.

2.2 Advantages of our proposed solution

1. Our system's distinctive advantages is its innovative, smartphone-optimized approach to video summarization. To the best of our knowledge, this is the first time a feature-rich video summarization system has been designed to run just on a smartphone. Through the use of improved processing algorithms for on-device execution and the advancements in mobile hardware capabilities, users may now benefit from the advantages of concise video summaries without relying on external servers or cloud-based processing.
2. Focusing on on-device processing is not merely a technology choice; it is a deliberate strategy to improve user privacy and lower server loads. Local video analysis on the smartphone reduces potential privacy risks by keeping sensitive data under the user's control. This is especially true for cloud-based solutions. By utilizing server resources more efficiently, this approach also contributes to lessening the strain on external servers and networks, increasing system sustainability and scalability.
3. The rate at which smartphone hardware is developing, particularly in terms of increased CPU and GPU capability, makes it increasingly feasible to use this powerful hardware. Completing complex video analysis tasks on-device is becoming both feasible and advantageous due to the increasing processing capability of modern smartphones. This local resource use not only increases the efficiency of video summarization but also adheres to the current trend of utilizing mobile devices as versatile, high-performance computing platforms.
4. One of the key benefits of this on-device video analysis system is its ability to blend in with the way customers interact with movies on their cellphones. Users are able to peruse, study, and comprehend shared flicks with ease using their devices. Users do not have to wait on distant servers to grasp the most crucial information in movies because of the system's local processing. This makes the experience quick and responsive. This technique encourages a more dynamic and fruitful interaction with mobile video content while accommodating the evolving expectations of consumers who need instant access to information.

3 Data Description

In our system, we use the CNN/DailyMail Dataset, which comprises 300,000 news stories and associated summaries from CNN and DailyMail Nallapati et al. (2016). This dataset is well-known for its summary capabilities. We also fine-tuned the GPT-2 model Radford et al. (2019) on this dataset to improve its summarization capabilities. We tested our model using a selection of 3-5 minute YouTube lecture video. This diverse collection, which reflects the breadth and depth of the dataset, assesses GPT-2's performance across a range of content types and durations in an attempt to validate its efficacy in generating accurate summaries in a number of real-world circumstances.

4 Methodology

In this section, we introduce our video summarization and analysis pipeline. Since videos contain audio and visual information, but commonly used language models primarily process text, we convert the video into a purely text based context representation. The audio is turned into a transcript via Vosk, a lightweight speech to text software for Android. Since we are focused on lecture slides, most visual content will be in the form of text on a screen. Optical Character Recognition (OCR)

Transformer enables the LLM to recognize relationships and connections

Figure 1: One of the Videos based on LLM's

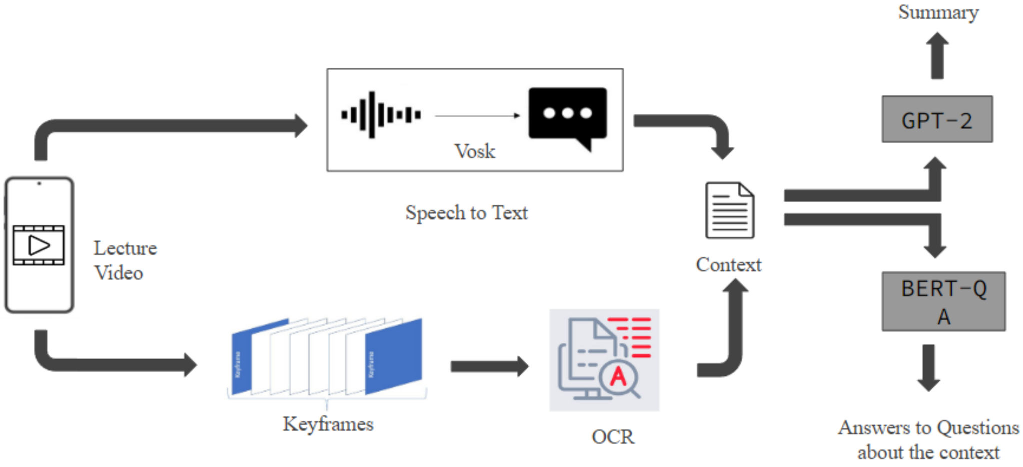


Figure 2: Overall System Workflow

is a simple solution to extract this information on relevant frames and combine it with the transcripts. To choose key frames to run OCR on, and extract the rich text information we develop a key frame sampling method. Once the context is fully in text form, we utilize BERT-QA for the question-and-answer task, and a fine-tuned version of GPT-2 for summarization.

1. **Android:** We build our app in Java on Android. Android is open source and has various useful libraries such as `mlkit` and `tflite` provided by Google. Additionally Android has 70.77% of the smartphone market share globally, and 41.64% of the US market share Afzal (2023), making it a very commonly used platform. Development for this project was done using Android Studio and tested on emulators and a physical Pixel 8 Pro smartphone. This phone with the Google Tensor G3 chipset is representative of the typical flagship android smartphone capable of running generative AI applications such as the pipeline we introduce in this paper.
2. **Audio Processing:** The Vosk Speech to Text API for Android is a versatile, lightweight, and offline-capable tool that is particularly useful for mobile applications requiring speech recognition functionality. Its open-source nature and support for multiple languages make it a practical choice for a wide range of applications. These attributes make it an ideal choice for our use case: transcribing the speech of lecture videos into text accurately. We choose the English-US model, as this is the language of most of the videos in our test set. The model runs very fast on the Pixel 8 Pro, taking 6 seconds to transcribe each minute of video. We found this API to be easy to integrate into our application. By adopting Vosk

for this application, we offer a swift and precise audio processing experience, which is essential to summarize videos in near real time for users.

3. **Key-Frame sampling:** In video analysis, choosing the right frames carefully for data extraction is essential. Processing all frames in a video is futile since consecutive frames would not have significant information gain. A simple idea is to perform uniform sampling of frames. Naïve sampling is not efficient since a large sample frequency will lead to redundant frames and a low frequency might lead to information loss. In order to address these issues, we implement a customized key-frame recognition algorithm. Unlike uniform sampling, we quantify the change in visual information of the frames using Frobenius Norm. We also use Pearson correlation coefficient as an additional filter to remove redundant frames. The Frobenius Norm may be used to rapidly assess the overall size, and the Pearson correlation coefficient can be used to more clearly identify the finer linear relationships with earlier frames. This two-pronged approach ensures a comprehensive understanding of deviations, allowing us to choose identify and extract frames that offer significant and unique insights.

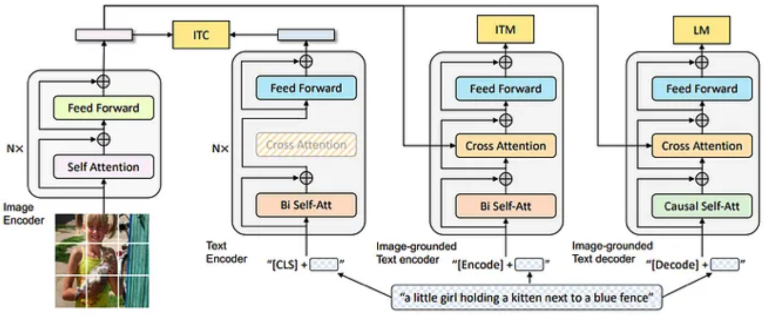


Figure 3: BLIP Architecture

4. **Image-data extraction:** Extraction of text information from visual data has been attempted in the form of image captioning. Image captioning models like BLIP Li et al. (2022) and GIT Wang et al. (2022) are state of the art approaches. These models have poor on-device execution time due to their huge size. In addition, they are trained on datasets like MS-COCO which makes them excellent at identifying humans and objects but they entirely skip text information on images, which is a core part in lecture videos. For instance, when we run the BLIP and GIT model on Fig 1, the caption generated is “a photograph of a woman with long hair and a red shirt”. The entire text information is discarded by these models. Thus, we chose the simplest and most effective way of Optical Character Recognition (OCR). OCR proved to be an excellent choice because of its fast execution and precision in recognising and transforming text from images, particularly in scenarios where text serves as the main content, such lecture videos and presentations.

5. **BERT QA:** BERT (Bidirectional Encoder Representations from Transformers) Devlin et al. (2019) is an important model in the field of natural language processing introduced by Google in 2018. It represents a significant shift in how machine learning models understand human language. Unlike previous models that processed text in one direction, either from left to right or right to left, BERT analyzes text in both directions simultaneously. This bidirectional approach allows BERT to capture the context of a word in a sentence more effectively, leading to a deeper understanding of language nuances.
Building on this foundation, the BERT-QA model specifically tailors the capabilities of BERT for Question Answering (QA) tasks. BERT-QA utilizes the powerful language understanding of BERT to analyze and understand both the question and the context in which the answer may lie. By processing the text bidirectionally, BERT-QA is able to discern the relationships and dependencies between words in a question and its potential answers, leading to more accurate and contextually relevant responses. This model delivers answers to user queries via extracting and returning the portion of text from the context that is most relevant to the question. We export this model to TensorFlow Lite, which is a version of

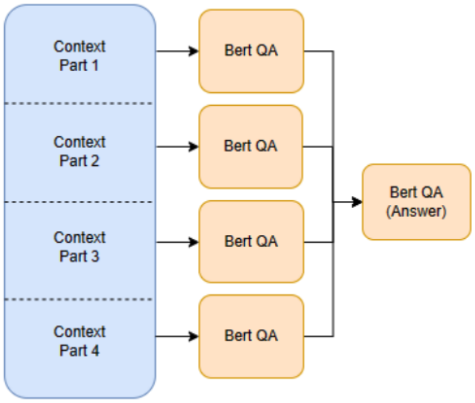
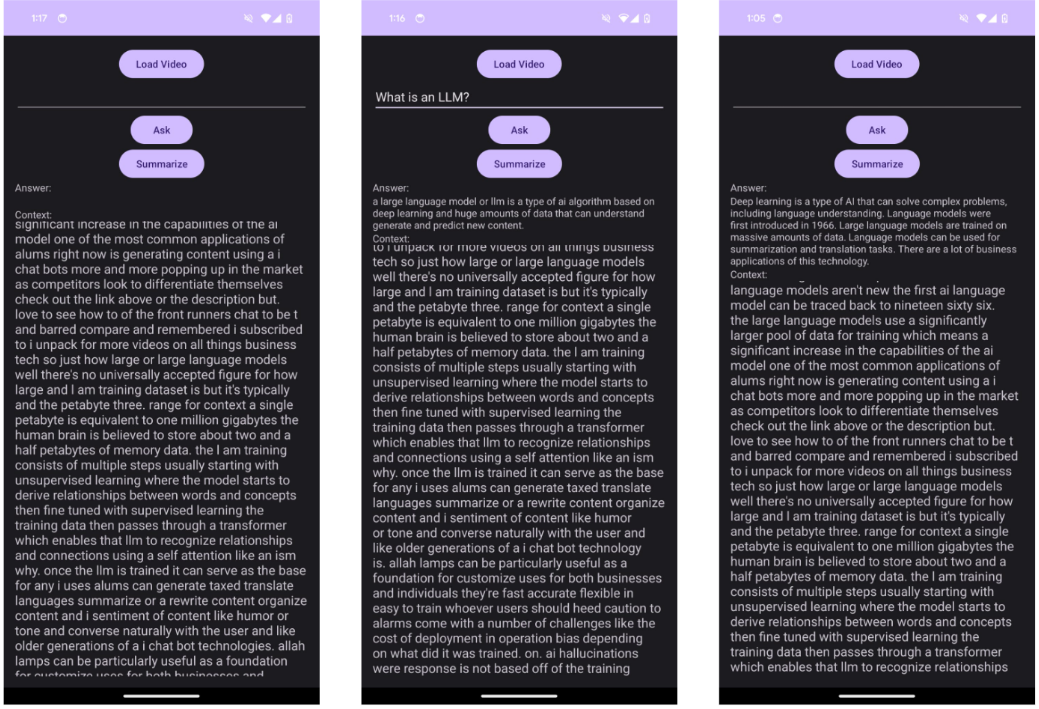


Figure 4: BERT-QA implemented on context length sized chunks of the text representation.

TensorFlow Abadi et al. (2015) that runs on Android devices.
When running this model as is, we experience the issue of limited context length. We are limited to 512 input tokens, where our context is often much longer for most videos. After exploring many potential solutions to this, we take a hierarchical approach of arriving at the final answer to a question. We first run BERT-QA over multiple context window sized chunks of the input and concatenate those results to form a new context. Then we run a BERT-QA call on this new context to pick the best answer of answers. In practice, this yielded the best answers to questions from the context, as the final run of BERT-QA measures relevancy of each answer candidate with the query. This approach improves the model's ability to process large amounts of contextual data by providing a reliable workaround for context length constraints.

6. **GPT:** Since we need to produce summaries, we need a generative language model that is powerful enough to process detailed contexts, but small enough to run on a smartphone. The smallest version of GPT-2, with 117 Million parameters fits these requirements.
GPT-2 Radford et al. (2019), which stands for Generative Pretrained Transformer 2, is an advanced language processing neural network developed by OpenAI. The model is trained on a vast corpus of text data, enabling it to understand and generate human-like text. One of the key features of GPT-2 is its ability to generate coherent and contextually relevant text over extended passages, showcasing a remarkable understanding of language nuances, styles, and even specific subject matter. However, this model is originally trained for the next token prediction task of completing likely next words that come in a paragraph.
Given the general nature of this model, it can be fine tuned to our summarization task as well. We use the CNN-Dailymail dataset for this. Because this dataset contains articles and one or two sentence summaries for them, we can arrange the fine-tuning set as "Article<TLDR>Summary" pairs i.e. we introduce the <TLDR>token to start the summarization task. For instance, the following summarizes a section of a narrative about an incident that happened aboard a cruise ship: "The ship's doctors say the elderly woman had hypertension and diabetes. Agencia Brasil reported that 86 passengers had already become ill while aboard." After a few epochs of fine tuning with this dataset, GPT-2 develops the ability to produce accurate summaries given the context a majority of the time. We use the same technique as we use with BERT-QA to address the context length issue, summarizing the article part by part, and using the concatenated result as our resulting summary.

7. **UI:** We created an Android application and included all the previously mentioned components. The novel aspect of our issue domain is our effort to develop an on-device video analysis system. Making a prototype using BERT-QA, GPT, Vosk, and key frame detection was our main objective. As a result, we created an intuitive Android application that showcases these features through a straightforward graphical user interface. By allowing users to upload movies and ask questions, the app showcases the smooth interaction on the screen.



Answer:

Context:
significant increase in the capabilities of the ai model one of the most common applications of alums right now is generating content using a i chat bots more and more popping up in the market as competitors look to differentiate themselves check out the link above or the description but. love to see how to of the front runners chat to be t and barred compare and remembered i subscribed to i unpack for more videos on all things business tech so just how large or large language models well there's no universally accepted figure for how large and I am training dataset is but it's typically and the petabyte three. range for context a single petabyte is equivalent to one million gigabytes the human brain is believed to store about two and a half petabytes of memory data. the I am training consists of multiple steps usually starting with unsupervised learning where the model starts to derive relationships between words and concepts then fine tuned with supervised learning the training data then passes through a transformer which enables that llm to recognize relationships and connections using a self attention like an ism why. once the llm is trained it can serve as the base for any i uses alums can generate taxed translate languages summarize or a rewrite content organize content and i sentiment of content like humor or tone and converse naturally with the user and like older generations of a i chat bot technologies. allah lamps can be particularly useful as a foundation for customize uses for both businesses and

What is an LLM?

Answer:
a large language model or llm is a type of ai algorithm based on deep learning and huge amounts of data that can understand generate and predict new content.

Context:
to i unpack for more videos on all things business tech so just how large or large language models well there's no universally accepted figure for how large and I am training dataset is but it's typically and the petabyte three. range for context a single petabyte is equivalent to one million gigabytes the human brain is believed to store about two and a half petabytes of memory data. the I am training consists of multiple steps usually starting with unsupervised learning where the model starts to derive relationships between words and concepts then fine tuned with supervised learning the training data then passes through a transformer which enables that llm to recognize relationships and connections using a self attention like an ism why. once the llm is trained it can serve as the base for any i uses alums can generate taxed translate languages summarize or a rewrite content organize content and i sentiment of content like humor or tone and converse naturally with the user and like older generations of a i chat bot technology is. allah lamps can be particularly useful as a foundation for customize uses for both businesses and individuals they're fast accurate flexible in easy to train whoever users should heed caution to alarms come with a number of challenges like the cost of deployment in operation bias depending on what did it was trained. on. ai hallucinations were response is not based off of the training

Answer:
Deep learning is a type of AI that can solve complex problems, including language understanding. Language models were first introduced in 1966. Large language models are trained on massive amounts of data. Language models can be used for summarization and translation tasks. There are a lot of business applications of this technology.

Context:
language models aren't new the first ai language model can be traced back to nineteen sixty six. the large language models use a significantly larger pool of data for training which means a significant increase in the capabilities of the ai model one of the most common applications of alums right now is generating content using a i chat bots more and more popping up in the market as competitors look to differentiate themselves check out the link above or the description but. love to see how to of the front runners chat to be t and barred compare and remembered i subscribed to i unpack for more videos on all things business tech so just how large or large language models well there's no universally accepted figure for how large and I am training dataset is but it's typically and the petabyte three. range for context a single petabyte is equivalent to one million gigabytes the human brain is believed to store about two and a half petabytes of memory data. the I am training consists of multiple steps usually starting with unsupervised learning where the model starts to derive relationships between words and concepts then fine tuned with supervised learning the training data then passes through a transformer which enables that llm to recognize relationships

Figure 5: First the user loads the video, and the extracted context appears in the scrollable text view (left). Then the user can ask questions (middle) or get a summary (right).

5 Results

We conducted a qualitative analysis to evaluate the quality of the content generated by GPT-2 (Summary) and BERT-QA (Answers to Questions). We created a survey where we posted a sample video and the summary generated by our model. We also added questions and responses generate by our on-device framework. We collected user responses on the accuracy of summarization and question answers. We used the Likert scale to measure if the generated summary and answers meet the evaluator's expectations. The video we surveyed is Eye on Tech (2023). We reproduce the questions asked in the survey:

1. **How accurate is this summary of the video?**
   Deep learning is a type of AI that can solve complex problems, including language understanding. Language models were first introduced in 1966. Large language models are trained on massive amounts of data. Language models can be used for summarization and translation tasks. There are a lot of business applications of this technology.
2. **How is this answer to: What is an LLM?**
   A large language model or LLM is a type of AI algorithm based on deep learning and huge amounts of data that can understand, generate, and predict new content.
3. **How is this answer to: What are the applications of LLMs?**
   One of the most common applications of LLM right now is generating content using ai chat bots. more and more of these are popping up on the market.
4. **How is this answer to: What are the challenges of LLMs?**
   A number of challenges like the cost of deployment and operational bias depending on what data it was trained on.

Table 1 shows the average rating for each of the four tasks described above. Our pipeline's performance was rated 4/5 or above for each of these tasks, showing that the information produced by this model seemed mostly accurate and effective to users who actually watched the full video.

Task #	Average User Rating
1	4.08
2	4.00
3	4.24
4	4.15

Table 1: Table showing survey results on the likert scale.

6 KEY TAKEAWAYS AND CONCLUSION

We have built and introduced a lightweight android app capable of combining audio and visual data in videos to summarize and answer questions about content on device. One advantage of this is increased privacy. Some videos may be under copyright or information protection and the use of third party services may not be allowed or advisible. Since this solution runs entirely on the smartphone, the video a user receives stays with the user. Another benefit is reduced network bandwith. Users on smartphones may have limited amounts of cellular data to use, and sending large videos to servers for processing would eat up a lot of bandwidth and potentially increase costs for the user. Our solution again gets around this issue by functioning with zero data use. Many apps and services that provide similar functions also have to manage the issue of server side scaling as the customer base increases. In this solution, customer acquisition will never be bottlenecked by an organization needing to increase hardware and software compute capabilities.

One limitation of this solution is the focus on lecture videos. As smartphone processors improve and image captioning models become more efficient, on-device solutions will become available to a wider range of video understanding tasks. Another future idea is the interpolation of image and audio data via attention, so that focus on different aspects of the video can be learned rather than using equal weightage of these input modes.

Overall, this is an exciting space, and we hope to see expanded uses of decentralized AI and on-device, especially for specialist purpose built models and preprocessing tasks.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Maleha Afzal. ios vs android market share by country: Top 30 countries using iphones, September 2023. URL `https://finance.yahoo.com/news/ios-vs-android-market-share-135251641.html`. Accessed 8 Dec 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

YouTube Eye on Tech. What is an llm (large language model)?, 2023. URL `www.youtube.com/watch?v=zKndCikg3R0`. Accessed 8 Dec 2023.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond, 2016.

Jeiyoon Park, Kiho Kwoun, Chanhee Lee, and Heuiseok Lim. Multimodal frame-scoring transformer for video summarization, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

David E. Rumelhart and James L. McClelland. *Learning Internal Representations by Error Propagation*, pp. 318–362. 1987.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language, 2022.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*. Springer, 2016.

Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. MM '17, pp. 863–871, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349062. doi: 10.1145/3123266.3123328. URL `https://doi.org/10.1145/3123266.3123328`.