AIDA: AI-POWERED MEDIA GENERATION

G. D. Bottari, P. Ariel

Globo, Brazil

ABSTRACT

Over the last decade, we have seen a steady demand for content on media platforms. However, producing quality content with attention to timing and relevance at scale is still a challenge. Simultaneously, developments in Artificial Intelligence (AI) research have enabled content generation such as text and voice with human-like quality for the first time.

In this work, we have automated the end-to-end production of a roundup newscast (a summary of current news). To achieve our goal, we have used AI and a life-like virtual character that delivers a news roundup. Our cloud-based system, named Aida, can either stream in real-time for traditional media channels and the web or generate videos for Video On Demand (VOD).

In the first stage of our pipeline, we used our in-house data-to-text Natural Language Generation (NLG) technology to describe the weather. We also built a text summarization engine to create short descriptions of full-length news articles automatically. After combining both texts into a script, we used text-to-speech based on deep-learning to create a natural-sounding voice with an emotional tone that matches the news content. The character's lips were audio-synced automatically and in real-time by analysing the previously generated audio. The resulting render can be customized to target different audiences by selecting news categories, virtual scenes, and even display relevant advertising.

INTRODUCTION

Workflow automation has been traditionally a remedy for repetitive and laborious tasks. From this perspective, automation is seen merely as a time-saving tool. In this work, we explore novel opportunities that open up beyond cost and time savings when we have an automated pipeline. Drawing upon different approaches for media generation, we discuss

opportunities and challenges on creating a news roundup program from scratch that is fully machine-generated.

Consuming automatically generated content is already pervasive in our modern daily lives, and it can come in different ways. Virtual assistants like Siri, Alexa, Cortana, and Google Assistant come to mind as people use them to ask for directions, order food, and other tasks. Many predict that this trend will continue as technology matures and fills new roles making people more reliant on their AI-powered assistants [1]. Surprisingly, for some applications, having a fully automated virtual human is preferred to having a human in the loop. That is the case of clinical interviews, where people are willing to be more open and relaxed when they believe the virtual human has no human operator oversight [2]. Also, a recent trend has virtual avatars gaining more space and acceptance on social networks, attracting millions of followers, and becoming advertising models for famous fashion brands [3].

That has led us to question if we could use a virtual human as a newscast anchor. By employing a virtual anchor, we can present a continuous information cycle that does not depend on human availability during business hours. Replacing anchors with virtual ones alone can cut production costs in half [4]. But perhaps, more interestingly, it opens up the possibility for unprecedented content personalization for deep user engagement.

The remainder of the paper is divided as follows. In the next session, we briefly discuss other related works and contextualize our approach. Next, we detail how our system works and our design decisions. In the next section, we discuss some opportunities and preliminary results that we gathered using our system. Finally, we conclude with some remarks and plans for continued exploration and evaluation of our work.

RELATED WORK

Advances on AI and Natural Language Generation over the past few years have enabled the development of Automated Journalism (AJ) tools. These tools help streamline the process of delivering news by organizing, interpreting, or presenting data to journalists. Some techniques even allow for the creation of news articles from data [5]. AJ tools allow journalists to focus on more complex and rewarding tasks while the computer can handle more mechanical work.

Speech synthesis through Text-to-speech (TTS) has also evolved substantially in the past few years. Traditional TTS approaches include concatenative and parametric methods [6]. The first method uses an extensive database of short sound bites and concatenates them to produce different speeches based on hard-coded rules for each text input. This system has the restriction of just producing limited content since it is manually programmed. On the other hand, parametric method models extract phonemes from the input text and pass hand-engineered parameters like fundamental frequency, magnitude, and spectrum into a vocoder, a set of algorithms that transform these data into human-like speech. Although more versatile, this approach produces a somewhat muffled and

robotic audio. Addressing both shortcomings, machine learning TTS approaches, particularly those using deep-learning [6], have enabled the development of engines that can generate natural-sounding speech from virtually any input text.

Thanks to the advancement of graphics hardware and software, today, it is possible to create realistic digital humans. When facing this challenge, there are usually two approaches: deepfake videos, which use deep learning systems to animate pictures and videos of real actors or traditional 3D modelling.

In the last decades, projects for the development of virtual anchors have emerged. Some works use techniques inspired by deepfake videos, like [7] and [8], that have created virtual newsreaders, based on machine learning, to deliver the news to viewers. The virtual news anchors' voices, lip movements, and facial expressions are inferred by the model that was previously trained with many hours of real footage.

Recent studies show that creating digital humans no longer requires time-consuming offline rendering methods. Harnessing the power of game engines, [9] and [10] have produced photorealistic characters using real-time rendering, motion capture, and facial capture.

Real-time rendering used by game engines enables the delivery of timely content that is currently not possible with deepfake methods. In addition to speed, it is often more versatile, because it enables easy switching of characters and their features such as clothes, hair, and makeup, and also modifying scene elements and camera.

OUR MEDIA GENERATION SYSTEM

As proof of concept, we developed a system where a user can create a newscast automatically by picking options from a webpage[1]. We designed the system to be simple for either a journalist or the end-user to operate.

Presentation tones

First, the user must choose what kind of tone to use with their target audience. In our system, we have two tones (or modes): one named “formal” while the other is labelled “informal” (see Figure 1). In the formal setting, Aida wears a sober outfit, has serious countenance, and talks with a neutral voice throughout the program. The scene depicts a classic newscast stand where she remains seated while a panel in the back displays illustrations relevant to the current news. On the other hand, in the informal setting, Aida is cheerful at the introduction when welcoming the user and even smiles. She stands beside a floating panel where the illustrations are displayed. Also, she wears a casual outfit and gesticulates more when talking. When designing both modes, we tried to target

[1] Some components we developed for this project are available at https://github.com/mediatechlab.

audiences with different preferences and tastes, making the user more comfortable and engaged.

Figure 1 – Presentation tones: (a) formal and (b) informal.

News selection

On another screen, the user checks out the news highlights for the week. These are provided by Bing News [11], which draws upon many different sources, like BBC, CNN, Fox News, and The New York Times, to name a few. Having a plethora of articles, we implemented a ranking algorithm to promote the ones that would better fit a newscast. Our algorithm uses some heuristics. Listed from the order of importance, we have:

- textstat's text standard[2]: measures the readability of a corpus based on its structure and is formed by a consensus of other metrics (most frequent value returned), such as Flesch-Kincaid Grade Level, Gunning FOG, SMOG Index, among others.
- presence of quotes: when Aida quotes from someone, it is easy to mistake their words for hers, so to avoid confusion, we rank down articles with many quotes.
- summary ratio: when summarizing long articles, there's an increased probability of extracting text that does not represent the article as a whole; therefore, we use this metric to rank down low summary ratios.
- first person use: as an editorial rule, we don't want Aida to reproduce opinion pieces, so that's why we rank down articles that write in that narrative voice.

[2] Please check https://github.com/shivam5992/textstat for the source code and a detailed explanation of this metric.

System architecture

When the user finishes selecting the most interesting headlines, the page sends their preferences and configurations to the aida-api module. This API orchestrates all the other services and interacts with the webpage on the tablet. A diagram for this architecture can be seen in Figure 2.

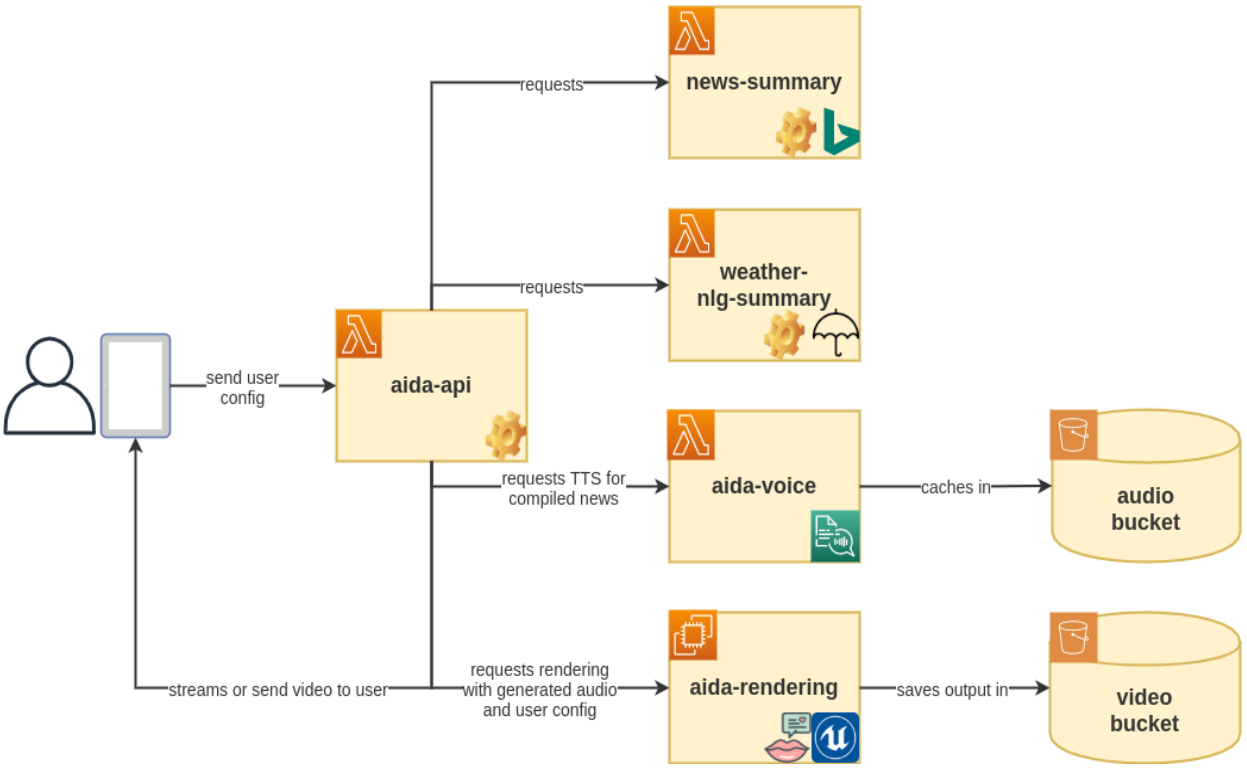

Figure 2 – System Architecture.

We use the URLs corresponding to the selected headlines to download (i.e., scrape) the full article. Once it has downloaded, we use a summarization algorithm to make the text shorter. We use an extractive summarization technique called LSA (Latent Semantic Analysis). It works by first calculating the term frequency (TF) of each sentence and storing it in a matrix A where each row is a unique word, and the indexes of the columns correspond to a sentence. LSA employs SVD (Singular Value Decomposition) on matrix A to effectively break down the original news piece into its concepts, represented by their base vectors, and the degree of importance of each concept represented by the singular value matrix. Following the algorithm described in [12], we can rank each sentence to their overall contribution to the document. Using this procedure, we may create a summary with as many sentences as desired. Pondering that the roundup newscast must

be brief, we set the limit to fit as many sentences (usually one or two) that don't go over 60 words.

For the next step, we generate natural-language text from weather forecast data provided by Dark Sky [13]. The technology we use is an in-house creation called AidaLang. With AidaLang, we implement a language-agnostic simple NLG pipeline divided into three input components. First, we have rule-based templates that can express sentences or entities in the text. To fill in the slots of the template, we have a section where data can be expressed freely in a document format. Lastly, a "storyline" component specifies which templates to use and with which data. The final product is output as natural language text by our algorithm.

Once we have generated both texts (news summaries and weather information), we use a Text-To-Speech (TTS) service from Microsoft [14] in the aida-voice module. It implements the so-called "neural" voices that are based on deep-learning and result in a realistic-sounding voice. Moreover, it allows the developer to specify different voice styles like "cheerful" and "empathetic." We take advantage of this to customize our "formal" and "informal" settings. After producing the speech files, we store them in an S3 bucket. This storage works like a cache to make the process faster and also to allow access to the audio files in the next step.

Finally, we can start the render process. We use a particular instance equipped with a Graphics Processing Unit (GPU) and a local webserver to orchestrate the process. When the server receives a request, it runs our application with Real-Time Messaging Protocol (RTMP) streaming enabled. This stream can be sent to the client directly, or we can save it locally to a file and later upload it to a video bucket. The stream is available in about 10 seconds, given that the rendering machine is already instantiated.

The rendering step, called aida-rendering in Figure 2, was developed on Unreal Engine 4 [15]. Game engines allow customization and insertion of new elements in real-time to the scene and, thanks to modern video cards, deliver high graphic quality. Some critical processes occur in runtime: changing the character clothes and the scenario to match the chosen mode (formal or informal); synchronizing the generated speech with the character's lip movements; loading facial expression animations that suit the tone of the presentation, and loading relevant illustrations for each news piece.

The lip-syncing used in this work is an adaptation of the tool provided by Oculus [16], which predicts the set of visemes (expressions of the lips and face that correspond to a particular speech segment) from the audio using a neural network. Such prediction is agnostic to the target language, although we only tested in English and Brazilian Portuguese. Each viseme is mapped to a specified geometry morph target with an expression strength. Moreover, the emotion is complementary to the lip sync, that is, we can activate facial expressions that make the character smile while speaking, for example.

When producing our virtual anchor, we followed the techniques outlined by [17] for the creation of digital humans: from the modelling process to the configurations of materials — using these techniques allowed both renderings in real-time with a high frame rate and with high graphic quality. Also essential to the process, were the animations that brought the character to life: they were captured by motion tracking with an actress and later refined with traditional animation.

DISCUSSION

In this section, we address the most relevant topics of our system: use cases and production costs, critiques about credibility, and our first experience with audience reception.

Use cases and production costs

Our first external demonstration was at IBC's 2019 Future Zone venue. That year we had a system much similar to what we have described here, but the rendering took place locally and not in the cloud. We collected mainly positive feedback and ideas. One idea, in particular, was to stream news from a website as they become available, creating a new broadcast channel. Others saw the potential to create a more personal message to its users, maybe even using different characters to appeal to each audience group.

Given different potential use cases, we analysed them regarding production costs for generating and storing video on the cloud. By that, we mean the cost to generate a new video, disregarding previous costs (i.e., 3D modelling and system development) and future distribution costs. See Table 1 for a summary of our findings.

The first use case, “3 min Roundups”, corresponds to a program as we described in the last session: a summary of current news events produced daily on weekdays for VOD consumption. With under two dollars cost per month, producing a show like this lowers the barrier of entry to many content producers that currently lack the means to produce a show traditionally besides representing extreme cost savings. However, as we have discussed earlier, we argue that the primary goal of such a system is to bring forth something that would not be possible before, namely, the customization of content. On “3 min Custom Roundups,” we estimated costs for producing 10 thousand videos that can represent customized news pieces, characters, settings, and tones for each user. In such a scenario, we have a total cost of USD 737.72, which gives about 7 cents per user. This use case represents deep customization of the generated content at a marginal cost.

Use case	Monthly Programs	Total Monthly Costs[3]
3 min Roundups	22	USD 1.62
3 min Custom Roundups	10,000	USD 737.72
20 min Newscasts	22	USD 10.82
24/7 News Channel	—	USD 389.84

Table 1 - Production costs.

Next, we projected the total cost for a 20-minute weekday newscast at USD 10.82. This kind of show has a higher video bitrate and can be used on traditional media channels. More extended programs like these can have diverse applications like presenting pre-recorded footage of news, giving the latest numbers during elections, or giving real-time commentary during game matches.

Similarly, we estimated that a continuous news channel would cost under $400 per month. We imagine that computer-generated content would not be enough to feed such a use case. However, during certain events like the Olympics or specific natural disasters, coverage naturally increases, and being able to deploy a temporary news channel could be invaluable.

Fake news and credibility

The main critique we received was about the credibility of having a robot delivering hard news in a world increasingly more concerned with fake news. We see the potential to harm and spread fake news this way. Especially now that texts can be generated with a level of quality that is indistinguishable from human-created ones in many cases. Even though we don't have a positive response to this, we believe that traditional news companies that have built their reputation can heavily influence consumer trust on new mediums like the one we are proposing here.

[3] We used the following information to produce a conservative estimate of costs. For the "Roundup" we estimate a 1080p@25fps H.264 (medium quality) video with 19.29 MB. The other use cases have a 39.59 MB estimate for H.264 MB with high quality. The main cost incurred is to keep the renderer running, since a g3s.xlarge instance costs USD 0.934 per hour. But we also added costs related to storage (S3) that is currently billed at USD 0.023 per GB per month and Microsoft's TTS engine which costs USD 16 per 1 million characters. We estimated that each video has 120 words or 564 characters per minute as we are not considering periods of silence and audio caching. Other costs were deemed negligible and omitted for clarity.

Audience reception

Back in Brazil, we had the opportunity to use the system with actual end-users for the first time during the latest Comic Con Experience (CCXP) in São Paulo. This venue is the largest pop culture festival in the world, with a total attendance of 280 thousand people gathering many content producers in the world. Globo had just gained exclusive rights in Brazil to a new show and was looking for a novel way to advertise it. This particular show is called Manifest, and the central premise is that a commercial aircraft vanished mysteriously, reappearing five and a half years into the future. When the plane lands, passengers and crew are confused because time has not passed from their point of view.

Figure 3 – A custom generated video for a test user.
The caption reads “flight 828 reappears after 5.5 years.”

Using the show's premise, we came up with a concept using Aida. First, a promoter from our staff collects some personal information like occupation, name, gender (so we know how to address them adequately), and we also take a picture if they consent to it. Using this information, we can transport the visitor to the show's story through a short video that is displayed on a TV. The visitor sees a breaking news announcement where Aida, the anchor, explains that the missing flight has just reappeared miraculously. Among the survivors, the visitor's picture is shown in a frame, and Aida uses their name and occupation during the report. Although the virtual newscast is being rendered in real-time, we also record it so we can send it to the user later, by e-mail and SMS, as a souvenir.

Even with a simpler scope than outlined here, we considered that the experience was very well received. Many visitors also shared their unique videos with family and friends on social media and others at the festival, which drove more people to the experience. After running for 5 days, we gathered 479 unique visitors, which we estimate as roughly 1 visitor every 5 minutes of the entire event length (40 hours). Considering the average length of the video, which was about 1 minute and 40 seconds, and the fact that we are

not considering downtime and lunch breaks, the experience gathered reasonable attention[4].

CONCLUSIONS & FUTURE WORKS

We have shown a framework for rapidly producing content and streaming it in real-time. We have implemented our vision as a roundup newscast with the latest news. This proof of concept represents attractive gains in cost and time. But more importantly, it unlocks new horizons to build personalized and targetable content at a scale that would never be possible using a traditional pipeline. Finally, we showed how scalable our personalized pipeline could be and some preliminary impressions of audience impact.

In the future, we believe in exploring other use cases for our system. We believe that sports, in particular e-sports, could provide a wealth of data to generate real-time analysis and commentary during matches. Data interpretation and content generation can adapt to each sport, enriching broadcasts with exclusive and accurate content about player's tactics, areas of expertise, the precision of specific actions, and other statistics that could not be analysed manually.

We also would like to measure the impact of selecting different characters or presentation modes in user engagement. We also think that it is worth exploring the difference in the impact of custom-made content compared with a broad and generic one. Following this study, we can also explore how personalized content synergizes with targeted ads.

ACKNOWLEDGEMENTS

We want to thank our colleagues for their support, encouragement, and feedback on all stages of the project.

[4] Watch a short making of video that features some automated content generated by our system here: https://youtu.be/O325AeUkwLU.

REFERENCES

1. Tulshan, A. S., Dhage, S. N., 2018. Survey on Virtual Assistant: Google Assistant, Siri, Cortana, Alexa. International Symposium on Signal Processing and Intelligent Recognition Systems (pp. 190-201). Springer, Singapore.

2. Lucas, G.M., Gratch, J., King, A. and Morency, L.P., 2014. It's only a computer: Virtual humans increase willingness to disclose. Computers in Human Behavior, 37, pp.94-100.

3. Fowler, D., 2018. The fascinating world of Instagram's 'virtual' celebrities. Accessed April 30, 2020 at https://www.bbc.com/worklife/article/20180402-the-fascinating-world-of-instagrams-virtual-celebrities.

4. Wiederhold, B.K., 2019. Animated News Anchors: Where to Next?. Cyberpsychology, behavior and social networking, 22(11), p.675.

5. Leppänen, L., Munezero, M., Granroth-Wilding, M. and Toivonen, H., 2017, September. Data-driven news generation for automated journalism. In Proceedings of the 10th International Conference on Natural Language Generation (pp. 188-197).

6. Tabet, Y. and Boughazi, M., 2011, May. Speech synthesis techniques. A survey. In International Workshop on Systems, Signal Processing and their Applications, WOSSPA (pp. 67-70). IEEE.

7. Brown, B. 2018. Virtual newscaster wants to know: Is this the real me or just a fantasy?. Accessed April 01, 2020 at https://www.digitaltrends.com/cool-tech/china-news-virtual-newsreader/.

8. Ivory, H. J. 2020. Reuters and Synthesia unveil AI prototype for automated video reports. Accessed February 12, 2020 at https://www.reuters.com/article/rpb-synthesia-prototype/reuters-and-synthesia-unveil-ai-prototype-for-automated-video-reports-idUSKBN2011O3.

9. Unreal Engine, 2019. Meet Vincent: a real-time digital human created in-house by a team of just five. Accessed April 20, 2020 at https://www.unrealengine.com/en-US/spotlights/meet-vincent-a-real-time-digital-human-created-in-house-by-a-team-of-just-five.

10. Cowley, D., 2018. Siren at FMX 2018: crossing the uncanny valley in real time. Accessed April 15, 2020 at https://www.unrealengine.com/en-US/events/siren-at-fmx-2018-crossing-the-uncanny-valley-in-real-time.

11. Microsoft Azure, 2019. Bing News Search. Accessed May 01, 2020 at https://azure.microsoft.com/en-us/services/cognitive-services/bing-news-search-api/.

12. Steinberger, J. and Jezek, K., 2004. Using latent semantic analysis in text summarization and summary evaluation. Proc. ISIM, 4, pp.93-100.

13. Dark Sky, 2019. Accessed March 14, 2020 at https://darksky.net/dev.

14. Microsoft Azure, 2020. Text to Speech. Accessed May 01, 2020 at https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech/.

15. Epic Games, 2020. Unreal Engine, Available at: https://www.unrealengine.com.

16. Oculus, 2018. Oculus Lipsync for Unreal Engine. Accessed April 29, 2020 at https://developer.oculus.com/documentation/unreal/audio-ovrlipsync-unreal/.

17. Epic Games, 2018. Digital Humans. Accessed April 30, 2020 at https://docs.unrealengine.com/en-US/Resources/Showcases/DigitalHumans/index.html.