US 20250265757A1

(19) **United States**

(12) **Patent Application Publication**

German et al.

(10) **Pub. No.: US 2025/0265757 A1**

(43) **Pub. Date: Aug. 21, 2025**

(54) **PREPROCESSOR SYSTEM FOR NATURAL LANGUAGE AVATARS**

(71) Applicant: **CodeBaby, Inc.**, Milwaukee, WI (US)

(72) Inventors: **Dan German**, Bristol (GB); **Michelle Collins**, Chicago, IL (US); **Tyler W. Chase-Nason**, Grand Junction, CO (US); **Navroz J. Daroga**, Franklin, WI (US)

(21) Appl. No.: **19/084,353**

(22) Filed: **Mar. 19, 2025**

Related U.S. Application Data

(63) Continuation-in-part of application No. 18/154,099, filed on Jan. 13, 2023, now Pat. No. 12,271,987.

(60) Provisional application No. 63/266,748, filed on Jan. 13, 2022.

Publication Classification

(51) **Int. Cl.**

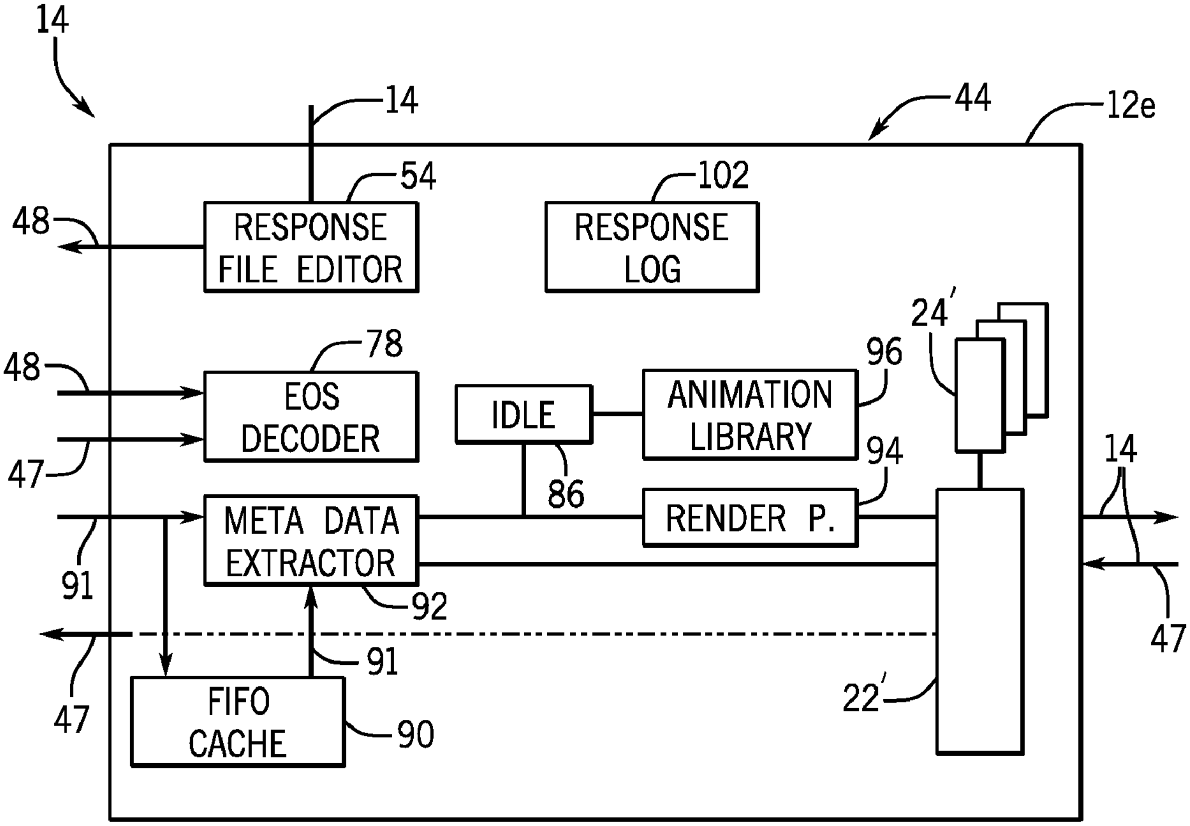
G06T 13/40	(2011.01)
G06F 16/9535	(2019.01)
G06F 40/30	(2020.01)
G06F 40/40	(2020.01)
G10L 13/08	(2013.01)

(52) **U.S. Cl.**

CPC ***G06T 13/40*** (2013.01); ***G06F 16/9535*** (2019.01); ***G06F 40/30*** (2020.01); ***G06F 40/40*** (2020.01); ***G10L 13/08*** (2013.01)

(57) **ABSTRACT**

A preprocessor for use with a machine learning system for control of computerized avatars provides for an embedding of avatar control information in a speech response file machine learning system for improved perception of emotional intelligence.



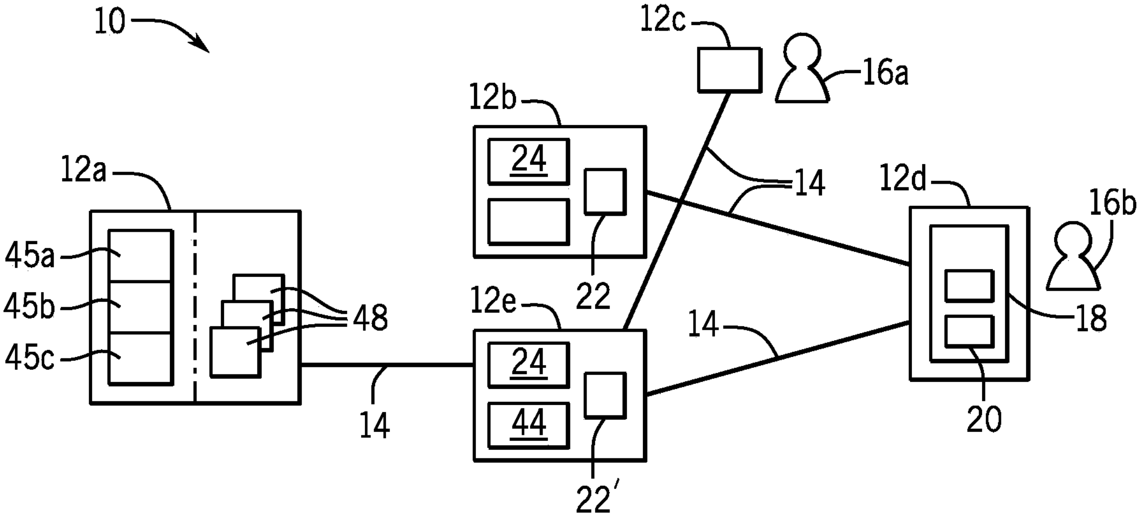

10
12c
16a
12b
24
14
12d
16b
12a
45a
45b
45c
48
22
12e
24
18
14
44
20
14
22′

FIG. 1

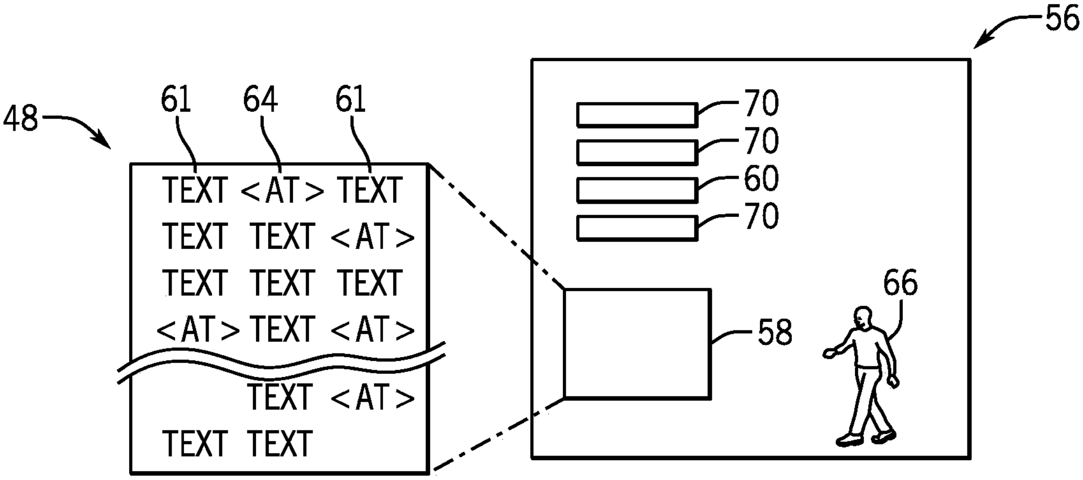

56
48
61
64
61
70
70
60
70
TEXT <AT> TEXT
TEXT TEXT <AT>
TEXT TEXT TEXT
<AT> TEXT <AT>
TEXT <AT>
TEXT TEXT
66
58

FIG. 2

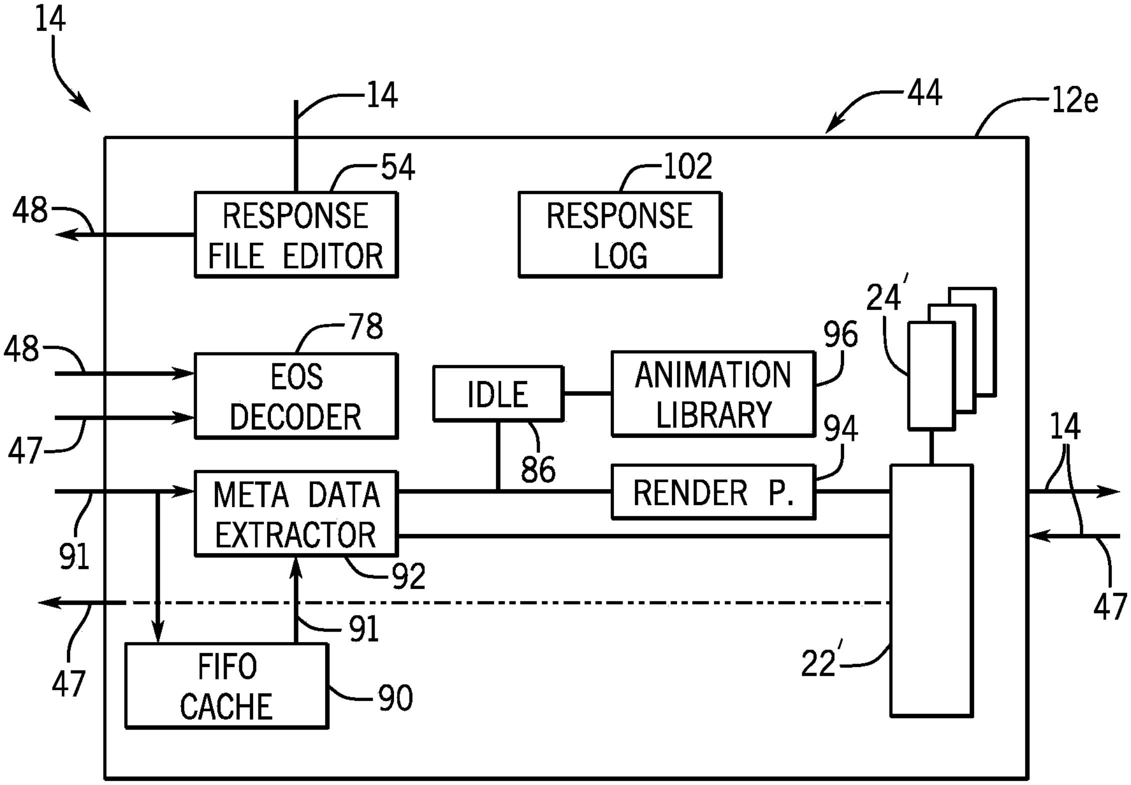

14
14
44
12e
54
102
RESPONSE FILE EDITOR
RESPONSE LOG
48
48
78
EOS DECODER
47
24′
96
IDLE
ANIMATION LIBRARY
94
86
14
META DATA EXTRACTOR
RENDER P.
91
92
47
91
47
FIFO CACHE
90
22′

FIG. 3

FIG. 4

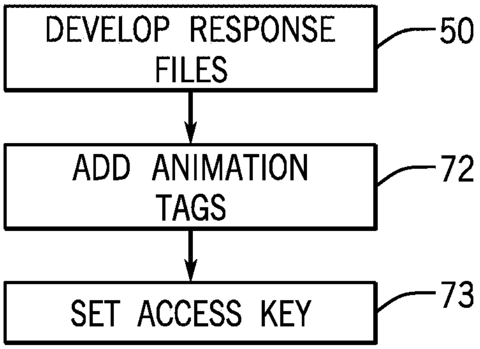
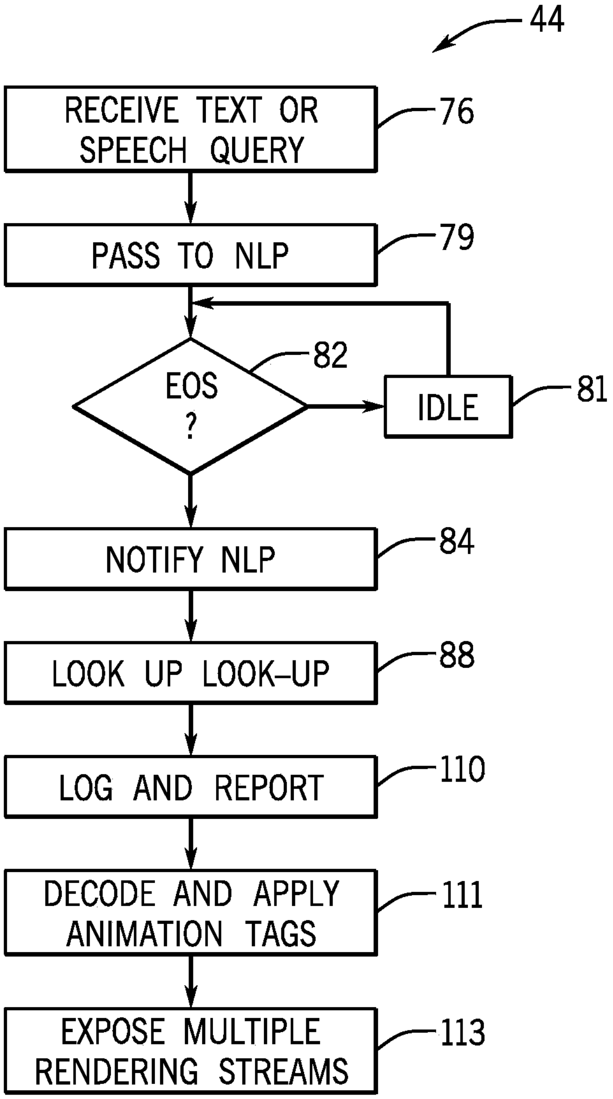

DEVELOP RESPONSE FILES
50
ADD ANIMATION TAGS
72
SET ACCESS KEY
73

FIG. 5


44
RECEIVE TEXT OR SPEECH QUERY
76
PASS TO NLP
79
EOS ?
82
IDLE
81
NOTIFY NLP
84
LOOK UP LOOK-UP
88
LOG AND REPORT
110
DECODE AND APPLY ANIMATION TAGS
111
EXPOSE MULTIPLE RENDERING STREAMS
113

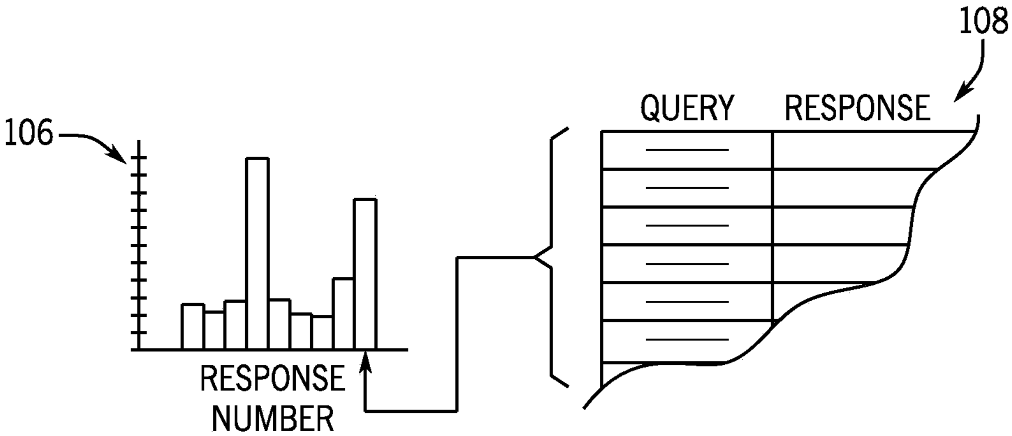

106
RESPONSE
NUMBER
QUERY
RESPONSE
108

FIG. 6

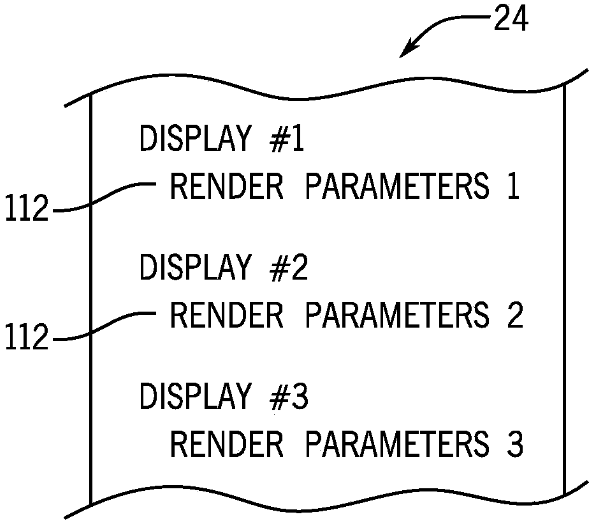

24
DISPLAY #1
RENDER PARAMETERS 1
112
DISPLAY #2
RENDER PARAMETERS 2
112
DISPLAY #3
RENDER PARAMETERS 3

FIG. 7

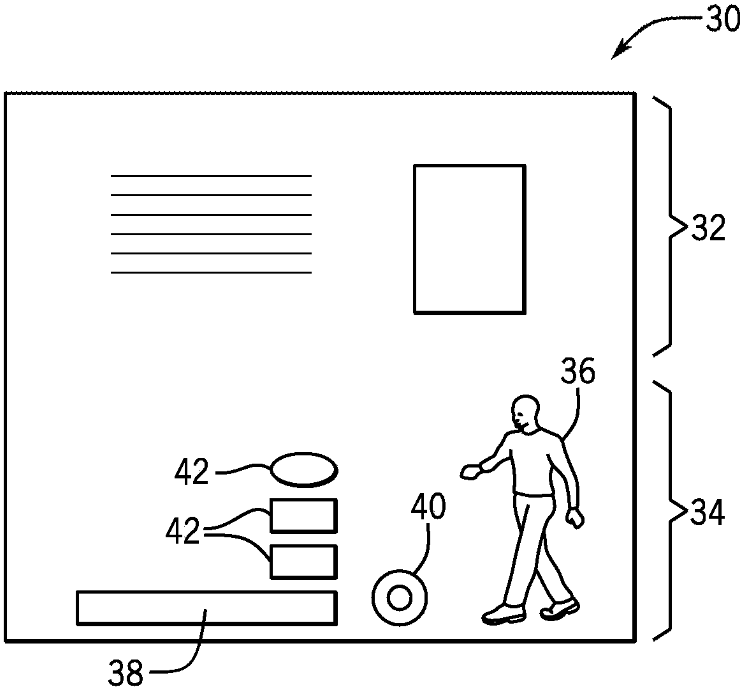


FIG. 8

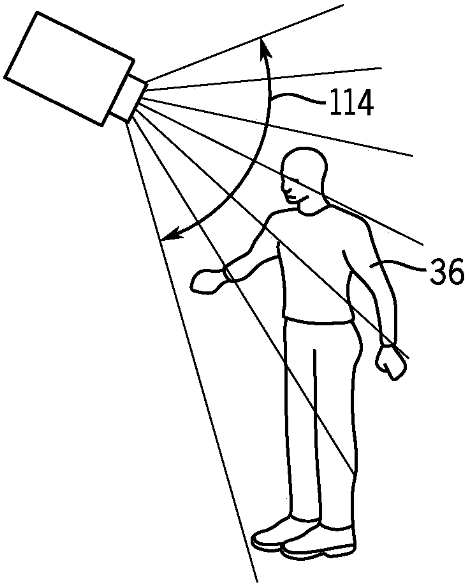


FIG. 9

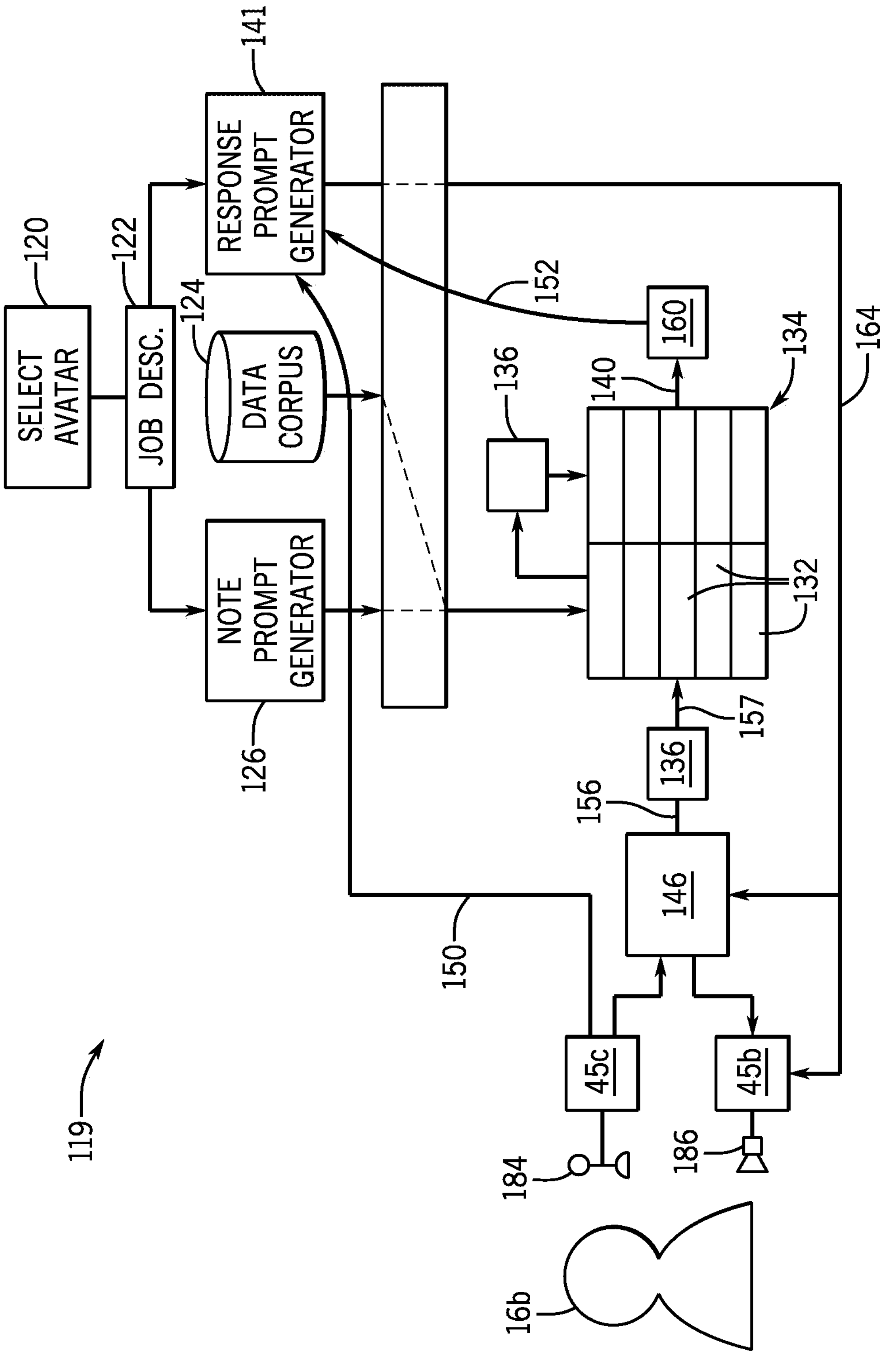

119
120
SELECT AVATAR
122
JOB DESC.
124
DATA CORPUS
126
NOTE PROMPT GENERATOR
141
RESPONSE PROMPT GENERATOR
152
160
136
140
134
132
164
157
136
156
146
150
45c
45b
184
186
16b

FIG. 10

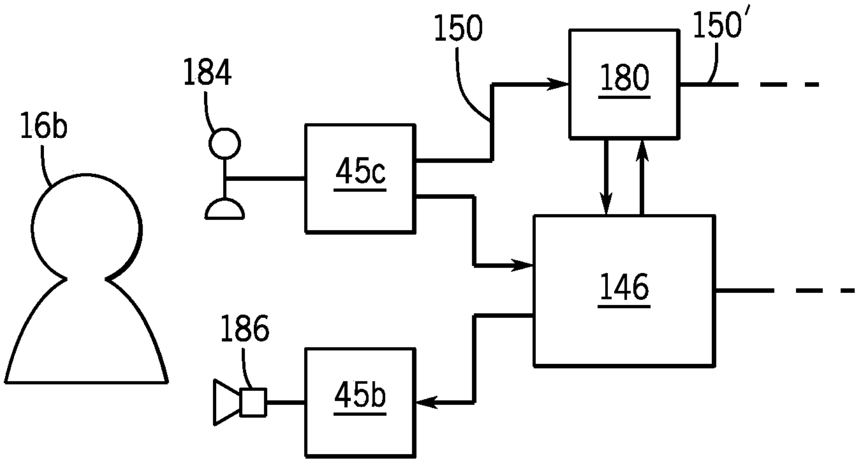


FIG. 11

PREPROCESSOR SYSTEM FOR NATURAL LANGUAGE AVATARS

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application is a continuation in part of U.S. application Ser. No. 18/154,099 filed Jan. 13, 2023, which claims the benefit of U.S. provisional application 63/266,748 filed Jan. 13, 2022, all hereby incorporated by reference.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] N/A

BACKGROUND OF THE INVENTION

[0003] The present invention relates to computer-generated avatars that can respond to natural language communications and, in particular, to a preprocessor positionable between a natural language engine and a consumer website improving the performance of such avatars.

[0004] Internet commerce and other on-line services struggle with the challenge of providing personalized service of a type normally associated with face-to-face interaction between individuals. To some degree, this challenge can be addressed using teams of people working at remote "call desks," or the like, to handle questions from service consumers; however, the solution has poor scalability when compared to a typical web service and can result in high costs and constrained levels of service.

[0005] In order to address this problem, computer-generated avatars have been developed providing animated representations of a human displayed on a computer screen and capable of interacting with consumers using natural language processing. Such avatars can be scaled with the Internet services they support, offering an economically sustainable high degree of service at low cost.

[0006] Unfortunately, the simulation of a human engaging in human interactions is difficult and avatar systems risk falling into the so-called "uncanny valley," a term capturing the observation that the closer one comes to simulating a human with an avatar, the more sensitive individuals interacting with the avatar are to off-putting flaws in that simulation.

SUMMARY OF THE INVENTION

[0007] The present invention helps create an avatar system that can better avoid the uncanny valley by improving avatar responsiveness and emotional intelligence. In one aspect, the invention allows the avatar animation to be precisely controlled with tags in the avatar responses, where the tags are generated by a prompt applied to a large language model preparing the response. By using the large language model, contextual gestures may be generated that are closer related to the responses and context.

[0008] These features are provided in a "preprocessor" which also simplifies the integration of a website to a natural language service provider. The interposition of this preprocessor allows important information that can be gleaned from the interaction between consumers and the avatars to be logged and captured for improving the avatar experience and aggregating this information even across different entities and avatar implementations.

[0009] More specifically, the present invention provides an avatar system using a large language machine model and a prompt generating system, the prompt generating system operating to: (1) receive a consumer query from a browser computer connected to the Internet; (2) generate a first prompt for the large language model based on the query, the first prompt including a preprepared avatar role description and instruction to identify speaker emotions related to the response; (3) apply the first prompt to a large language model to generate a response package including response text and speaker emotions; and (4) convert the text of the response package to speech together with avatar control data interspersed in the speech at times based on locations of the avatar control data tags with respect to the text of the response package for output at the browser computer.

[0010] It is thus a feature of at least one embodiment of the invention to tightly integrate avatar animations with responses developed by large language models to improve the perceived emotional intelligence of the avatar.

[0011] The machine learning system may further operate to generate a second prompt for the large language model, the second prompt including the preprepared avatar role description and instructions to review identified digital data to prepare a set of notes based on that data relevant to fulfilling the preprepared avatar role description. This second prompt may then be applied a to the large language model to generate a set of notes that are referred to in the first prompt.

[0012] It is thus a feature of at least one embodiment of the invention to allow the large language model to be automatically and closely tailored to a particular job responsibility, for example, as a spokesperson for company or the like.

[0013] The prompt generating system may further operate to process the notes to develop associated semantic addresses for each note and to collect a history of previous queries from the customer to develop at least one semantic address from the history. This one semantic address is then used to identify the specific notes used in the first prompt.

[0014] It is thus a feature of at least one embodiment of the invention to tailor content informing the large language model to the recent history of queries.

[0015] The prompt generating system may further generate a third prompt for the large language model, the third prompt referencing a collected history of previous queries from the customer and providing instructions to identify current queries according to a speaker of the query to discount queries that are an echo of the response text generated by the large language model.

[0016] It is thus a feature of at least one embodiment of the invention to prevent responses from the large language model from feeding back into the information of the queries such as could corrupt the responses, for example, a consumer microphone picking up output audio and interpreting it as if it were said by the consumer. It is yet another feature of at least one embodiment of the invention to block this feedback in the presence of a noisy or corrupted communication channel between a speaker and microphone used by the consumer.

[0017] The avatar system may further include a preprocessor receiving the query from the browser computer and forwarding it to the large language model and receiving the speech and avatar control data from the large language model and forwarding it to the browser computer.

[0018] It is thus a feature of at least one embodiment of the invention to provide a preprocessor allowing these features to be added to current commercial natural language processing systems.

[0019] In some cases, the avatar system may include a website processor communicating website data to the browser computer, the website data including a script directing the browser computer to the preprocessor to provide a query.

[0020] It is thus a feature of at least one embodiment of the invention to permit the use of a commercial large language model for a variety of websites that do not intrinsically support such capabilities.

[0021] The preprocessor may include an animation table indexed by the avatar control data to generate animation commands effecting a predetermined avatar model animation on the browser computer.

[0022] It is thus a feature of at least one embodiment of the invention to provide more sophisticated avatar control while working with current natural language processors by supporting an independent avatar animation system.

[0023] The preprocessor animation table may include animations linked to visemes, and the preprocessor may process the speech control data to identify visemes and timing of the visemes to control an animation using the animation table.

[0024] It is thus a feature of at least one embodiment of the invention to provide dynamic lip-synching based on text of the response allowing accurate synchronization for changing, for example, personalized portions of the response that cannot be readily prerendered.

[0025] These particular objects and advantages may apply to only some embodiments falling within the claims and thus do not define the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] FIG. **1** is a network diagram showing an embodiment of the preprocessor of the present invention positioned between a service-consumer browser and one or more web-based natural language services, the preprocessor operating in parallel with a provider web server allowing simple integration of these computational resources;

[0027] FIG. **2** is a screen display of an editor running on the preprocessor of FIG. **1** for linking preprepared animations to avatar response packages to improve the emotional intelligence of an avatar and further showing a simplified representation of an annotated response file;

[0028] FIG. **3** is a detailed block diagram of the avatar preprocessor of FIG. **1** showing multiple components implemented as tasks on one or more computers;

[0029] FIG. **4** is a flowchart showing a general workflow for configuring the avatar preprocessor of FIG. **1**;

[0030] FIG. **5** is a flowchart of a program executable on the avatar preprocessor of FIG. **1** to implement the depicted multiple components;

[0031] FIG. **6** is an example report that can be obtained from the data logging step of FIG. **5** possible with the preprocessor configuration;

[0032] FIG. **7** is an example web page that may be served by the preprocessor of FIG. **3** providing different rendering options;

[0033] FIG. **8** is an example depiction of a web page as rendered on a browser showing a superimposed avatar and various avatar communication controls;

[0034] FIG. **9** is a simplified perspective view of a virtual camera controllable in a 3-D rendering system for real-time rendering of the avatar;

[0035] FIG. **10** is a block diagram of a second embodiment of the invention using a large language model; and

[0036] FIG. **11** is a fragmentary view of an additional speaker identification system enabling interruption response.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0037] Referring now to FIG. **1**, an avatar system **10** may be formed of one or more network-interconnected computers including, for example, one or more natural language services computers **12***a* implementing a natural language processor, a service-provider (henceforth "provider") computer **12***b* operating as a Web server, an interface computer **12***c*, a service-consumer (henceforth "consumer") computer **12***d*, and a preprocessor computer **12***e*.

[0038] Each of these computers **12** may communicate, for example, over a computer network such as the Internet or within a cloud server as indicated by connecting lines **14** representing communication media, switches, routers, etc., as is understood in the art. Individually, the computers **12** each may provide standard computer architectures including one or more processors, network interfaces, and computer memory such as random-access memory and nonvolatile storage elements such as disk drives or the like that can hold stored programs and data as will be described.

[0039] It will be understood that this is a simplified representation and that an avatar system **10** contemplates multiple consumer computers **12***d* and consumers **16***b*, for example, connecting simultaneously to the avatar system **10** which may develop multiple instances of the various components that will be described according to techniques known in the art.

[0040] The interface computer **12***c* and the consumer computer **12***d* will normally provide human interface devices such as a graphic display screen, a keyboard, and the like to allow bi-directional communication with a human producer **16***a* (for example, representing an Internet-based business or the like) and a human consumer **16***b* (for example, representing an individual purchasing an item or seeking information on the web). In addition, each of the consumer computer **12***d* and the interface computer **12***c* may include, in memory, a web browser **18**, for example, being an executable program based on Chrome, an open source web browser managed by Google (a subsidiary of Alphabet) of Mountain View, California. The browser **18** provides a web-based interface between the preprocessor computer **12***e* and either or both of the interface computer **12***c* and the consumer computer **12***d*. Generally, the browser **18** on the consumer computer **12***d* will include a rendering engine **20**, for example, WebGL™, an open source rendering engine managed by Khronos Group, Inc. of Beaverton, Oregon, USA. The rendering engine **20** will work with the hardware (including graphic processing units) of these consumer computers **12***d* to render in real time two-dimensional projections of three-dimensional animated models as will be described.

[0041] The provider computer **12***b* will generally provide a Web server **22**, for example, the Apache open-source Web server, managed by the Apache Software Foundation, and will further include software-implementing services useful to the consumer **16***b* such as for the sale of goods or the provision of information or the like. In this regard, the provider computer **12***b* may hold in memory one or more web pages **24** providing display instructions to web browsers **18** (for example, using protocols such as HTML and CSS known in the art). The web browser **18** of the consumer computer **12***d* may display information on a graphic display **30** associated with the consumer computer **12***d* generally related to that service as display portion **32**. The web pages **24** may reference an underlying database engine (not shown) and stored media files (not shown), including music and images that may be referenced and thus served by the web pages **24** to be displayed or presented at the consumer computer **12***d*.

[0042] The service provided by the provider computer **12***b* may benefit from a natural language avatar to assist the consumer **16***b* in various tasks conducted through the Web server **22**. For example, the avatar may be used to explain how to use those services or to provide additional information about the services or products being sold. Referring also to FIG. **8**, for this purpose, the preprocessor computer **12***e* may also provide a web server **22'** holding one or more web pages **24'** operating to serve to the consumer computer **12***d* the necessary information to implement a natural language avatar interface in a display portion **34**. This display portion **34** is shown sharply demarcated from display portion **32**; however, in practice these display portions may overlap, for example, with the avatar system being superimposed on the underlying web page of display portion **32**.

[0043] Generally, the display portion **34** provided by the preprocessor computer **12***e* may implement the avatar system including, for example, an animated avatar **36** providing a rendering of a human figure, a text entry box **38** allowing natural language queries to be entered as text, a microphone activation button **40** allowing natural language queries to be entered as speech, and one or more preset questions or action buttons **42** that may invoke responses from the avatar system when activated by the consumer **16***b*.

[0044] This integration of web page information from two different computers, the producer computer **12***b*, and the preprocessor computer **12***e*, is preferably implemented by adding a single script instruction to the web page **24** of producer computer **12***b* being viewed by the consumer **16***b* which redirects the browser **18** of the computer **12***d* to the preprocessor computer **12***e* for the portion **34**, as is generally understood in the art.

[0045] More generally, the preprocessor computer **12***e* provides a bi-directional interface between the consumer computer **12***d* and the one or more natural language services computers **12***a*, the latter of which include software components providing services including a natural language engine **45***a* (interpreting natural language phrases as mapped to intents), a text-to-speech converter **45***b* (converting ASCII text into audio speech files), and a speech-to-text converter **45***c* (converting audio speech files into ASCII text). In one example, natural language engine **45***a* and speech-to-text converter **45***c* may be implemented by Google's Dialogflow virtual agent available from Google (Alphabet), cited above, and the text-to-speech converter **45***b* necessary to provide the avatar system with an audio output may be provided by Amazon Polly, commercially available from Amazon Web Services of Seattle, Washington, USA. The text-to-speech conversion may provide for timing signals and lip synchronization signals for a rendered avatar as well as speech in an audio streaming format.

[0046] In one embodiment, the lip synchronization may be done dynamically using information of the text-to-speech conversion which produces not only an audio data stream but also phonemes and phoneme timing signals. In this regard, the present invention may map the phonemes to visemes using a preprepared mapping table (the visemes providing appropriate lip animations for the phoneme) which in turn may be linked to prerendered animation components for the particular visemes. To the extent that the animation data may be stored on the browser **18**, this greatly reduces the amount of data that needs to be sent to the browser **18** and further allows dynamically generated text content to be properly synchronized to the avatar's lips, for example, as is necessary with text that is personalized to a particular user experience and thus not known in advance as would permit pre-rendering.

[0047] During operation, the preprocessor computer **12***e* will execute a program **44** to receive text or speech from the consumer **16***b* interacting with display portion **34** and will forward that information to the natural language services computers **12***a*. The natural language services computers **12** will then convert any speech to text using the speech-to-text converter **45***c* and will provide that text to the natural language engine **45***a* and to the preprocessor computer **12***c*. The natural language engine **45***a* will then identify the received text phrases to particular intents, the intent generally abstracting the purpose of the consumer's query and will map those intents to one of multiple text response objects **48**. The text response objects **48** will typically have been prepared by the producer **16***a* using an interface computer **12***c* communicating with preprocessor computer **12***e* as will be discussed below.

[0048] In an alternative embodiment the function of the natural language engine **45***a* and the text response objects **48** are implemented through a large language model that may be trained as discussed below. This large language model then receives text from the consumer **16***b* to directly provide response text.

[0049] Once a text response object **48** is identified, if the text response object **48** has not previously been cached as will be discussed below, the associated text of that text response object **48** is converted to an audio speech file **91** which is sent together with timing information and lip synchronization information to the preprocessor computer **12***c*. This timing and lip synchronization information (as well as the animation tags to be described below) are embedded as metadata in the audio stream to be closely synchronized with that audio.

[0050] In turn, the preprocessor computer **12***e* serves the speech files **91** to the browser **18** of the consumer computer **12***d* and uses the metadata of the lip synchronization signals and animation tags to provide rendering information to the browser **18** of the consumer computer **12***d* necessary to animate a rendering of the avatar **36** in time with the speech files **91**.

[0051] Importantly, queries and questions from the consumer **16***b* and consumer computer **12***d* and responses to those questions all pass through the preprocessor computer **12***e* allowing single point monitoring and logging of this information that can be used both to improve the natural language processing and to provide insights into end-user behavior and the like. Multiple instances of the components of the preprocessor computer **12***e* (for example, associated with different consumers **16***b* on different consumer computers **12***d*) may provide data to a common log to provide a broader understanding of effectiveness and possible improvements to the natural language processing as well as important insights into questions held by consumers and the like.

[0052] Referring now to FIGS. **3** and **4**, a key part of this avatar system **10** of FIG. **1** is developing the text response objects **48** and their mappings to intents. As indicated by process block **50** of FIG. **4**, for example, these response objects **48** may be prepared by the producer **16***a* using an interface computer **12***c* communicating with preprocessor computer **12***e* and an editor module **54** implemented through a web interface with the preprocessor computer **12***c*.

Embodiment I

[0053] Referring now also to FIG. **2**, the editor module **54** may provide an editor screen **56** (viewable on the browser **18** of the interface computer **12***c*) presenting a text entry block **58** into which a response text **61** for a response object **48** may be entered and linked to a particular intent. Intents may be generated by the natural language engine **45***a*, for example, reviewing live chats and performing a clustering analysis. For example, the natural language engine **45***a* may indicate an intent being a question about color, and the response may be the text **61** of "Would you like me to show you some color samples?" This text **61** is entered into a text box of the editor screen **56**, for example, by typing.

[0054] The present invention further allows this text **61** of a response object **48** to be annotated with animation tags **64** located within the text **61**, for example, between words, using conventional editor commands. The particular animation tag **64** may be selected from an animation tag menu **60** per process block **72** of FIG. **4**. This menu **60** may identify a set of preprepared animation scripts **96** (shown in FIG. **3**) having instructions that can be interpreted by the rendering engine **20** of a browser **18** in rendering an avatar **36** to perform specific animations. For example, a particular animation script **96** may have the rendered avatar **36** look left, or up, or right, or down, or smile, or lean inward, or look concerned, or point with one or both hands or rub eyes or bend down to pick up a dropped object or other similar activities that suggest emotional empathy. These tags **64** (which may be expressed with a unique ASCII delimiter sequence) are then embedded into the text **61** at the point where the animation should be initiated. The animation tags **64** may more broadly provide for other web automation including changing web pages, displaying, or playing other media resources, activating anchor buttons, changing focus, filling form fields, and the like. When an animation tag **64** is selected from the menu **60**, the associated animation script **96** may be played on an example rendering of an avatar **66** identical to the ultimately produced avatar **36** allowing the results of the animation script **96** to be reviewed. Similarly, the entire response object **48**, including multiple animation tags **64**, may be played for review. The response file object **48** may also include voice tags such as rate, tone or volume and other metadata such as clickable responses and page redirects which may be implemented in a separate payload section.

[0055] In all cases, the editor screen **56** may also provide for other selections **70**, for example, allowing the particular avatar **36** to be tailored for particular genders or races and for the synthesized speech (by text-to-speech converter **45***b*) to provide different voices or do different voice tones (happy, sad, etc.). These selections may also be embedded in the text **61** in the form of voice or expression tags (not shown). The completed text response object **48** may then be uploaded to the natural language engine **45***a* as linked to a particular intent.

[0056] Referring again to FIG. **4**, as a final step in the configuration process, per process block **73**, the producer **16***a* may provide an access code to the preprocessor computer **12***e* allowing preprocessor computer **12***e* to receive and provide communications to natural language services computers **12***a*.

[0057] Referring now to FIGS. **3** and **5**, once the configuration steps of FIG. **4** are completed, the preprocessor computer **12***e* may operate, as indicated by process block **76**, to receive text or speech expressing a query **47** from the consumer **16***b* through browser **18** on consumer computer **12***d* which may be passed to the natural language engine **45***a* (and in the case of speech-to-text conversion, to the speech-to-text converter **45***c*) of the natural language services computers **12***a* per process block **79**. During any speech-to-text conversion by the speech-to-text converter **45***c*, text of a decoded query **47** is also provided to preprocessor computer **12***e* indicating in run-time each decoded word (subject to later correction). This information is processed by an end-of-speech (EOS) detector **78** in preprocessor computer **12***e* as indicated by decision block **82**.

[0058] The EOS detector **78** monitors the decoded text of the query **47** to determine an end-of-speech by the consumer **16***b* indicating completion of a thought or sentence. The natural language engine **45***a* also independently attempts to determine end-of-speech, normally by monitoring spectral energy in the voice band; however, the EOS detector **78** attempts to anticipate the end-of-speech determination of the natural language engine **45***a*, for example, by monitoring a delay after a last successfully decoded word by the speech-to-text converter **45***c*. As one example, a timer may be reset every time someone speaks. When that timer finally gets to a predetermined value, it is assumed that there is an end-of-speech and the response may be generated. This EOS detector **78** is less sensitive to in-band noise which might prolong natural language processing when no further information has been conveyed. This improved sensitivity of the EOS detector **78** may improve responsiveness of the avatar system and thus the inferred empathy by the avatar to queries by the consumer **16***b*.

[0059] If end-of-speech is detected by the EOS detector **78** before the natural language engine **45***a*, the natural language engine **45***a* is notified per process block **85** and the natural language engine **45***a* promptly outputs a determined intent identifying a response object **48**. The response object **48** is sent to the text-to-speech converter **45***b* together with all of the tags related to voice selection, tone, etc., which are used to control the resulting synthesized speech file **91**. The response object **48** is also forwarded by the EOS detector **78** to a response log **102** to be described below.

[0060] Until an end-of-speech is detected, the program **44** loops through an idle block **81** waiting for the end-of-speech to be determined either by the EOS detector **78** or the natural language engine **45***a*. At idle block **81** and at all idle times, idle animations are provided to the browser **18** animating the avatar **36** to mimic the natural dynamic movements of a waiting but attentive individual, for example, a shifting of position, blinking, a tipping of the avatar's head or the like. These idle animations may be preprogrammed (like the animation scripts **96**) and may be generated at random times randomly by an idle engine **86** (shown in FIG. **3**).

[0061] At process block **88**, following the processing of an end-of-speech detection per process block **84** but prior to receiving speech file **91** from the text-to-speech converter **45***b*, the preprocessor computer **12***e* reviews a response cache **90** to see if an audio speech file **91** (and its associated metadata) is currently held in the response cache **90** for the particular text processed by the text-to-speech converter **45***b* and previously obtained from the speech-to-text converter **45***c*. The response cache **90** may, for example, may be a first-in, first-out cache indexed by the associated text of the text response object **48** (or a hash or similar indexing scheme). If the speech file **91** is found in the response cache **90**, the avatar **36** may respond more quickly (before the generation of speech files **91** by the text-to-speech converter **45***b*) without the delay associated with text-to-speech conversion by using the cache-stored audio data. If the necessary speech files **91** are not found in the response cache **90**, the program **44** waits for the speech files **91** from the text-to-speech converter **45***b* which are then added to the response cache **90** indexed by the associated the text response object **48**. Without waiting for the caching process, the rendered speech **91** is used in animation of the avatar **36** as provided to an audio player of the browser **18** of consumer computer **12***d*.

[0062] The speech file **91** is first provided to a parser **92** which can extract the metadata from the speech file (lip movement and animation tags and timing) and is then output to the consumer computer **12***d* via server **22'** per process block **111**. Simultaneously, the metadata, including timing information including lip movement and animation tags **64**, are extracted and forwarded to a rendering instruction generator **94** which develops commands for the lip shapes and for the animation tags **64** and also provides these rendering instructions to the Web server **22'** (or as noted below, to the browser **18** if offloaded). In this regard, the rendering instruction generator **94** receives the animation tags **64** passed from the parser **92** and uses them to index through a table of pre-rendered animations scripts **96** previously linked to the text response objects **48** as described above with respect to FIG. **2**. The animation information is served to the browser **18** of consumer computer **12***d* in time with the streamed speech file from the cache **90** as a dynamic content portion of the web page **24'** where it controls the rendering engine **20**. By employing human-curated animation effects tightly linked to a text file and intent, appropriate emotional intelligence is displayed by the avatar **36** during performance.

[0063] For reasons of efficiency, the parser **92**, idle engine **86**, animation scripts **96**, end of speech detector **78**, and rendering instruction generator **94** may be offloaded as a program to be executed by the browser **18** of the consumer computer **12***d*. In this respect, this offloaded program should be considered part of the processor computer **12***c*.

[0064] Referring now to FIGS. **3**, **5** and **6**, as noted above, response log **102** may be maintained on the preprocessor computer **12***e* by the preprocessor program **44**, for example, capturing the text of the query **47** received by the preprocessor computer **12***e* and the text of the text response objects **48** provided to consumer computer **12***d* after conversion to speech. This log information may be optionally used to inform the cache **90** by having a retention policy that treats favorably more common responses. More generally, however, this log information may be used to provide useful information to optimize the text response objects **48** to respond to common queries **47** and to provide market intelligence relating to concerns by consumers **16***b*. In this regard, the program **44** may develop a dashboard (for example, visible on an interface computer **12***c* or similar computer) providing, for example, a histogram **106** indicating the frequency of particular responses of text response objects **48** over a predetermined time. A histogram bar may be selected to reveal a table **108** providing the actual query **47** and the corresponding text of the associated text response objects **48**. In this way, common queries **47** may be quickly identified, for example, to indicate end-user questions allowing the web page **24** to be optimized to eliminate ambiguity and/or to improve the training of the natural language engine **45***a* or to develop additional responses for these particular queries that may provide better resolution and hence better consumer experience. Importantly, this information may also be aggregated anonymously with other instances of the preprocessor program **44** to provide a more comprehensive view of the natural language processing success to allow it to be further refined over time. This logging and generation is shown, for example, in FIG. **5** as process block **110**.

[0065] Referring now to FIGS. **7** and **9**, in process block **113** of FIG. **5**, a high degree of responsiveness of the avatar **66** may be provided for a wide range of consumer computers **12***d* having different processing capabilities, for example, by providing on the web page **24** a set of conditional rendering parameters **112** associated, for example, with different processor capabilities or different display sizes of the consumer computer **12***d*. This conditional rendering can be implemented, for example, using the conventions of cascading style sheets (CSS) which allow different formats of a web page to be displayed on different displays. Importantly, the conditional rendering parameters **112** may control the rendering window **114** of the model of the avatar **36**, generally indicating the area of the model being depicted and the angle of depiction. For example, for use with higher powered computers **12***d* and larger displays, the camera parameters may be set to capture within the field of view of the rendering window **114** the entire avatar **36** whereas for use with lower powered computers with smaller displays (for example, cellular phones) the camera parameters, such as the rendering window **114**, may be set to show only the head and shoulders of the avatar **36**. The effect of this is both to preserve screen area on a smaller screen but to also greatly reduce the rendering time and thus increase battery power and necessary processing overhead of portable devices such as cell phones or the like. It will be appreciated that the camera angle may also be adjusted according to capabilities of the consumer computer **12***d*, for example, to provide the avatar **36** with the general top view when the avatar **36** is located near the bottom of a small display or frontal view when the avatar **36** is located in full figure to the side of the display. Rendering lighting, frame rate, and other effects may also be changed to emphasize the avatar on screens of lower or higher resolution.

Embodiment II

[0066] Referring now to FIGS. **1** and **10**, a variation of the above-described system may employ a large language model

(LLM **118**) and a prompt generating system **119** as an alternative to using preprepared text response objects **48** as described above. Instead, in this embodiment, the LLM **118**, implemented on the natural language services computers **12*a***, may generate response objects directly from received text or speech per process block **76** of FIG. **5**.

[0067] In this embodiment, the producer **16*a*** using the interface computer **12*c*** may communicate with the preprocessor computer **12*e*** to select a style of an avatar **66** (shown, for example, in FIG. **2**) per process block **120**, for example, being a selection among different cartoon-like characters whose visual characteristics (gender, race, etc.) vary.

[0068] As indicated by process block **122**, the producer **16*a*** may then compose and enter a job description characterizing the role of the avatar **66** within a given organization. This job description may be entered as a text file, for example, as follows:

[0069] You are a sales and support Avatar on an organization's website.

[0070] You represent an organization called Company A.

[0071] Your name is Bob.

[0072] Next, a data corpus **124** is identified, for example, consisting of text documents that will provide the LLM **118** with context about Company A. Such documents may include but not be limited to a link to the website of Company A and other documents describing Company A taken from standard company literature or prepared for this purpose. Importantly, this information includes information that would be needed by a support person in replying to questions.

[0073] An example LLM **118** suitable for use with this purpose is GPT-4 from OpenAI, although many LLMs released after 2023 can serve this function.

[0074] The job description prepared at process block **122** is then provided to a note prompt engine **126** serving to create a text prompt for the LLM **118**, for example, by concatenating information of the job description into a larger prompt, for example, as shown below:

[0075] In the role of a customer service representative for Company A, review the corpus data to extract a set of note summaries of information that would be in your capacity for helping customers.

[0076] The LLM **118** receives the data corpus **124** and the prompt from the note prompt engine **126** to generate a set of "notes" **132** extracted from the data corpus **124** using retrieval augmented generation (RAG). RAG is a common industry process that utilizes an embedding algorithm to convert text into a series of numbers representing the content and context of the text. These sets of numbers can quickly be compared with other sets of near matching text. The closest matching text based on the number set is given to the LLM to help guide and inform its response. These notes **132** are related to the avatar role at Company A and used to populate a note database **134**.

[0077] The notes **132** may then be analyzed semantically using a semantic processor **136** to produce a semantic address **140** for each note **132**, the semantic address **140** being a multidimensional numeric vector reflecting the substantive content of the note **132**. A nonlimiting example semantic processor **136** suitable for use for this purpose is text-embedding-3-small from OpenAI.

[0078] Referring still to FIG. **10**, the job description prepared at process block **122** is also used by a response prompt generator **141** to generate a prompt for the LLM **118** related to current queries by a consumer **16*b*** as discussed above. As before, the consumer query may be converted into text by a speech-to-text converter **45*c*** and provided as current query input **150** to the response prompt generator **141**. In addition, a note data input **152** is provided to the response prompt generator **141**.

[0079] This note data input **152** is generated from recent history of the chat which is collected as a text file in chat history buffer **146**. This chat history buffer **146** is used to extract a chat history on multiple time scales including, for example, the last response by the consumer **16*b***, the last five responses by the consumer **16*b***, and the entire conversation with the consumer **16*b*** to that point. This extracted data **156** is then provided to the semantic processor **136** to extract a query semantic address **157**.

[0080] The query semantic address **157** may be matched to the semantic addresses **140** of the note database **134** to identify a set of notes **132** relevant to the current transaction with the consumer **16*b***. The information of these notes **132** is collected in a response note buffer **160** until it the desired critical mass of notes is developed, the size being determined empirically. The contents of the response note buffer **160** and, as noted above, the job description from process block **122** and the most recent query from the consumer **16*b***, together inform the response prompt generator **141** in generating a prompt to the LLM **118** to elicit an avatar response **164**. The avatar response **164** is then provided to a text-to-speech processor **45*b*** providing an audio output to the consumer **16*b*** and updating the chat history buffer **146**.

[0081] The response prompt generator **141** may use a template that also instructs the LLM **118** to automatically generate the animation tags **64** to be embedded in the response as described above. For this purpose, the pool of prepared animation scripts **96** may be associated with emotional states such as interest, amusement, concern, boredom and the like and the LLM prompt instructed to consider the likely emotional state of itself as a speaker and to embed corresponding tags **64** in its text output (marked to be ignored by the text-to-speech processor but captured by the rendering instruction generator **94** to be discussed below). Similar tags **64** may be used to control the tone of the voice (happy, sad, etc.) captured by the text-to-speech converter.

[0082] When giving a response please consider the tone of the response and add appropriate markup, for both voice and avatar animations.

To Change the Avatar Animations You can Use Markup in the Form Below:

```
<mark name='{"emotion":"Happy"}'>This makes me Happy</mark>
<mark name='{"emotion":"Unsure"}'>Humm, I'm really not sure</mark>
<mark name='{"emotion":"Cheer"}'>Hurray we won</mark>
<mark name='{"emotion":"Sassy"}'>Well wouldn't you like to know!</mark>
<mark name='{"emotion":"Grateful"}'>Thank you so much!</mark>
<mark name='{"emotion":"Angry"}'>This sort of thing makes me furious</mark>
```

-continued

```
<mark name='{"emotion":"Annoyed"}'>I just don't want to talk about it</mark>
<mark name='{"emotion":"Silly"}'>I can't stop giggling, this is so silly!</mark>
<mark name='{"emotion":"Excited"}'>I can't wait to create a new CodeBaby avatar!</mark>
<mark name='{"emotion":"Kind"}'>Always here to help.</mark>
<mark name='{"emotion":"Fitness"}'>Time to crush this workout.</mark>
<mark name='{"emotion":"Wave"}'>Hi there, how are you?</mark>
```

To Alter the Rendered Audio Speech You can Use the Following Markup:

```
<prosody pitch="x-high">I am speaking in an extra high pitch</prosody>
<prosody pitch="high">I am speaking in a high pitch</prosody>
<prosody pitch="medium">I am speaking in a medium pitch</prosody>
<prosody pitch="low">I am speaking in a low pitch</prosody>
<prosody pitch="x-low">I am speaking in an extra low pitch</prosody>
<prosody volume="x-soft">I am speaking with extra soft volume</prosody>
<prosody volume="soft">I am speaking with soft volume</prosody>
<prosody volume="medium">I am speaking with medium volume</prosody>
<prosody volume="loud">I am speaking with loud volume</prosody>
<prosody volume="x-loud">I am speaking with extra loud volume</prosody>
<prosody rate="x-slow">I am speaking extra slowly</prosody>
<prosody rate="slow">I am speaking slowly</prosody>
<prosody rate="medium">I am speaking normally</prosody>
<prosody rate="fast">I am speaking fast</prosody>
<prosody rate="x-fast">I am speaking extra fast</prosody>
I am going to put a <emphasis level="strong">strong</emphasis> on the word strong.
I am going to put a <emphasis level="moderate">moderate</emphasis> on the word moderate.
I am going to put a <emphasis level="reduced">reduced</emphasis> on the word reduced.
<mstts:express-as style="customerservice">How can I help you today?</mstts:express-as>
It's easy as <say-as interpret-as="digits">1 2 3</say-as>
The correct way to spell pineapple is <say-as interpret-as="spell-out">pineapple</say-as>
10 + 10 is <say-as interpret-as="number">20</say-as>
He came in <say-as interpret-as="ordinal">1</say-as> place
About <say-as interpret-as="fraction">1⁄12</say-as> of all men are colorblind
Today's date is <say-as interpret-as="date">1/1/2025</say-as>
It is <say-as interpret-as="time">1:00PM</say-as>
You can reach my cell at <say-as interpret-as="telephone">630-222-2222</say-as>
The Willis Tower is located at<say-as interpret-as="address">233 S Wacker Dr, Chicago, IL
60606</say-as>
<say-as interpret-as="expletive">censor this</say-as>
This is a <break strength="x-weak"></break>extra weak break
This is a <break strength="weak"></break> weak break
This is a <break strength="medium"></break> medium break
This is a <break strength="strong"></break> strong break
This is a <break strength="x-strong"></break> weak break
This is a <break time="3000ms"></break> 3 second break
An american might say<lang xml:lang="es-US">howdy partner</lang> if they were from the
south.
It has a certain <lang xml:lang="fr-FR">je ne sais quoi</lang> don't you think
<lang xml:lang="en-CA"></lang>
In Britain they might say <lang xml:lang="en-GB">Jolly good old chap</lang> but they sound
<emphasis level="strong">very</emphasis> silly.
In India, they might say <lang xml:lang="en-IN">Namaste, yaar!</lang>
In Spain, they might say <lang xml:lang="es-ES">¡Hola, tío!</lang>
In Australia, they might say <lang xml:lang="en-AU">G'day, mate!</lang>
In Germany, they might say <lang xml:lang="de-DE">Hallo, Kumpel!</lang>
In Brazil, they might say <lang xml:lang="pt-BR">E aí, parceiro?</lang>
In Canada, they might say <lang xml:lang="fr-CA">Salut, mon chum!</lang>
In Mexico, they might say <lang xml:lang="es-MX">¿Qué onda, compa?</lang>
In Japan, they might say <lang xml:lang="ja-JP">やあ、元気</lang>
In Italy, they might say <lang xml:lang="it-IT">Ciao, amico!</lang>
```

[0083] Referring now to FIG. **11**, the current query input **150** from the speech-to-text converter **45***c* may be further provided to a speaker identification generator **180** identifying the received text to particular speakers using diarization available from a number of services including Google cited above. The diarized speech may then be processed to identify any apparent speaker that is in fact a likely echo of a current response generated by the LLM **118** passing out of speaker **186** and being picked up by the microphone **184**. This identification of LLM speech can be performed by a similarity matching between historical diarized speech and known speech generated by the LLM **118** and held in the chat history buffer **146** to match one diarized speaker to the LLM **118**. The similarity matching can be used to clearly identify the responses by the LLM **118** in the current query input **150** to produce tagged input **150'** informing the LLM **118** that some of the received text is simply an echo. The LLM **118** may be instructed in the job description prompt to disregard such echos.

[0084] The speaker identification generator **180** may also identify cases where multiple speakers are speaking simultaneously such as suggest an interruption by the consumer of the output from the LLM **118**. Such interruptions are identified by a change of speaker from the LLM **118** to another speaker during an output by the LLM **118**. This interruption information may be provided to a gate **151** blocking the remainder of the current response by the LLM **118** (that is being interrupted) and flushing the remainder of the current response being prepared by the LLM **118** to initiate preparation of a new response based on the interrupting information allowing a more natural interaction with the consumer **16***b* In this way, the microphone **184** may be constantly active allowing the consumer **16***b* to interrupt the response process by the large language model **118** and for this interruption to be detected.

[0085] Other features of this embodiment may follow the teaching of the Embodiment I discussed previously.

[0086] Certain terminology is used herein for purposes of reference only, and thus is not intended to be limiting. For example, terms such as "upper", "lower", "above", and "below" refer to directions in the drawings to which reference is made. Terms such as "front", "back", "rear", "bottom", and "side", describe the orientation of portions of the component within a consistent but arbitrary frame of reference which is made clear by reference to the text and the associated drawings describing the component under discussion. Such terminology may include the words specifically mentioned above, derivatives thereof, and words of similar import. Similarly, the terms "first", "second" and other such numerical terms referring to structures do not imply a sequence or order unless clearly indicated by the context.

[0087] When introducing elements or features of the present disclosure and the exemplary embodiments, the articles "a", "an", "the" and "said" are intended to mean that there are one or more of such elements or features. The terms "comprising", "including", and "having" are intended to be inclusive and mean that there may be additional elements or features other than those specifically noted. It is further to be understood that the method steps, processes, and operations described herein are not to be construed as necessarily requiring their performance in the particular order discussed or illustrated, unless specifically identified as an order of performance. It is also to be understood that additional or alternative steps may be employed.

[0088] While the programs used to provide the services and functions described above have been described with respect to particular computers and locations for clarity, it will be understood that the present invention is inherently distributed allowing programs and their services and functions to be flexibly relocated among different computers and thus for the extent of the computers to be flexibly defined over multiple discrete machines.

[0089] It is specifically intended that the present invention not be limited to the embodiments and illustrations contained herein and the claims should be understood to include modified forms of those embodiments including portions of the embodiments and combinations of elements of different embodiments as come within the scope of the following claims. All of the publications described herein, including patents and non-patent publications, are hereby incorporated herein by reference in their entireties

[0090] To aid the Patent Office and any readers of any patent issued on this application in interpreting the claims appended hereto, applicants wish to note that they do not intend any of the appended claims or claim elements to invoke 35 U.S.C. 112(f) unless the words "means for" or "step for" are explicitly used in the particular claim.

We claim:

1. An avatar system comprising:

a large language learning model; and

a prompt generating system operating to:

(1) receive a consumer query from a browser computer connected to the Internet;

(2) generate a first prompt for a large language model based on the query, the first prompt including a preprepared avatar role description and instruction to identify speaker emotions related to a response;

(3) apply the first prompt to a large language model to generate a response package including response text and speaker emotions; and

(4) convert the text of the response package to speech together with avatar control data interspersed in the speech at times based on locations of the avatar control data tags with respect to the text of the response package for output at the browser computer.

2. The avatar system of claim **1** wherein the prompt generating system further operates to:

generate a second prompt for the large language model, the second prompt including the preprepared avatar role description and instructions to review identified digital data to prepare a set of notes based on that data relevant to fulfilling the preprepared avatar role description;

apply the second prompt to the large language model to generate a set of notes; and

wherein the first prompt includes a reference to specific notes.

3. The avatar system of claim **2** wherein the digital data includes a website of a company, and the preprepared avatar role description is that of customer service representative for the company.

4. The avatar system of claim **2** wherein the prompt generating system further operates to:

process the notes to develop associated semantic addresses for each note;

collect a history of previous queries from the consumer to develop at least one semantic address from the history; and

use the at least one semantic address to identify a specific note for use in the first prompt.

5. The avatar system of claim **1** further including a referencing of a collected history of previous queries from the consumer and identify current queries according to a speaker of the query to discount queries that are an echo of the response text generated by the large language model.

6. The avatar system of claim **5** wherein the avatar system continuously monitors responses from the customer and detects responses that interrupt an outputting of response text to the customer to cease that response text and prepare a new first prompt based on the interruption.

7. The avatar system of claim **1** further including a preprocessor receiving the query from the browser computer and forwarding it to the large language model and receiving the speech and avatar control data from the large language model and forwarding it to the browser computer.

8. The avatar system of claim **7** further including a website processor communicating website data to the browser computer, the website data including a script directing the browser computer to the preprocessor to provide a query.

9. The avatar system of claim **7** wherein the preprocessor includes an animation table indexed by the avatar control data to generate animation commands effecting a predetermined avatar model animation on the browser computer.

10. The avatar system of claim **9** wherein the preprocessor animation table includes animations linked to visemes and the preprocessor processes the speech control data to identify visemes and timing of the visemes to control an animation using the animation table.

11. A method of implementing an avatar using a large language model comprising:

(1) receiving a consumer query from a browser computer connected to the Internet;

(2) generating a first prompt for a large language model based on the query, the first prompt including a prepared avatar role description and instruction to identify speaker emotions related to a response;

(3) applying the first prompt to a large language model to generate a response package including response text and speaker emotions; and

(4) converting the text of the response package to speech together with avatar control data interspersed in the speech at times based on locations of the avatar control data tags with respect to the text of the response package for output at the browser computer.

* * * * *