



- (51) **International Patent Classification:**
G06N 20/00 (2019.01) *G06Q 10/10* (2023.01)
- (21) **International Application Number:**
PCT/US2023/012627
- (22) **International Filing Date:**
08 February 2023 (08.02.2023)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
17/667,470 08 February 2022 (08.02.2022) US
17/690,424 09 March 2022 (09.03.2022) US
17/690,451 09 March 2022 (09.03.2022) US
- (71) **Applicant: MY JOB MATCHER, INC. d/b/a JOB.COM**
[US/US]; 108 WILD BASIN ROAD, STE 250, AUSTIN,
Texas 78746 (US).
- (72) **Inventors: STEWART, Arran;** 108 WILD BASIN
ROAD, STE 250, AUSTIN, Texas 78746 (US). **O'BRIEN,**

Steve; 108 WILD BASIN ROAD, STE 250, RALEIGH,
Texas 78746 (US).

(74) **Agent: DRAYTON, Micah;** 200 Clarendon Street, 59th
Floor, Boston, Massachusetts 02116 (US).

(81) **Designated States** (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE,
KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU,
LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG,
NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS,
RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH,
TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS,
ZA, ZM, ZW.

(84) **Designated States** (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, CV,
GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,
TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,

(54) **Title:** APPARATUS AND METHOD FOR AUTOMATED VIDEO RECORD GENERATION

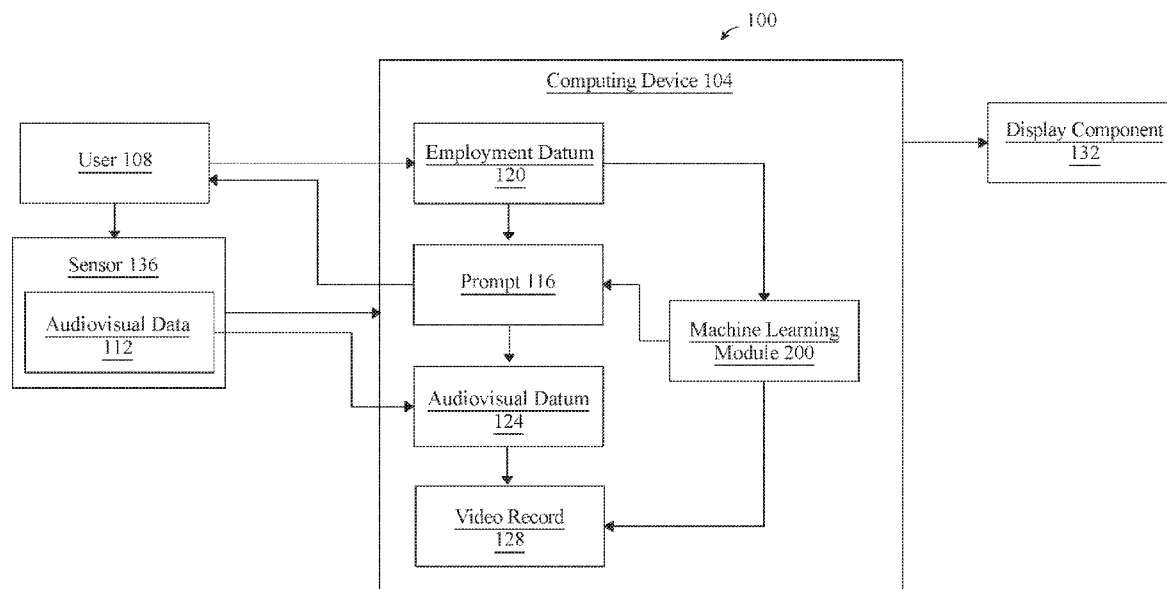


FIG. 1

(57) **Abstract:** A method for automatic video record generation is provided. Method may include a plurality of sensors configured to detect at least audiovisual data. Method may include a processor that may be configured to generated prompts for a user to respond to. User responses may be collected and formed into a video record. Method may use machine learning to automatically generate video records by assembling a plurality of temporal sections gathered from audiovisual data from a user.

TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

APPARATUS AND METHOD FOR AUTOMATED VIDEO RECORD GENERATION CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of priority of U.S. Nonprovisional Application Serial No. 17/667,470, filed on February 8, 2022, and entitled “APPARATUS AND METHOD FOR
5 AUTOMATED VIDEO RECORD GENERATION,” U.S. Nonprovisional Application Serial No. 17/690,451, filed on March 9, 2022, and entitled “APPARATUS AND METHODS FOR CREATING A VIDEO RECORD,” U.S. Nonprovisional Application Serial No. 17/582,087 filed on January 24, 2022, and entitled “DIGITAL POSTING MATCH RECOMMENDATION APPARATUS AND METHODS”, U.S. Nonprovisional Application Serial No. 17/582,070, filed
10 on January 24, 2022, and entitled “SYSTEMS AND METHODS FOR PARSING AND COMPARING VIDEO RECORD DUPLICATIONS”, U.S. Nonprovisional Application Serial No. 17/582,059 filed on January 24, 2022, and entitled “APPARATUS AND METHODS FOR MATCHING VIDEO RECORDS WITH POSTINGS,” U.S. Nonprovisional Application Serial No. 17/690,424 filed on March 9, 2022, and entitled “APPARATUSES AND METHODS FOR
15 LINKING POSTING DATA”, which is incorporated herein by reference in its entirety all of which are incorporated by reference herein in their entirety.

FIELD OF THE INVENTION

The present invention generally relates to the field of human resource technology. In particular, the present invention is directed to apparatuses and methods for automated video record
20 generation.

BACKGROUND

Video content (including short-form video content) has steadily been rising in popularity for many years. Its advantages as a communication medium are manifest. However, present ways of doing business are not capable of making optimal use of this form of media.
25

SUMMARY OF THE DISCLOSURE

In an aspect an apparatus for automated video record generation includes a plurality of sensors configured to: capture at least audiovisual data from a user; generate at least an audiovisual datum based on the at least audiovisual data from a user; at least a processor
30 communicatively connected to the plurality of sensors; and a memory communicatively connected to the processor, the memory containing instructions configuring the at least a

processor to: convert the at least an audiovisual datum into a video file; and generate a video record of a user based on the at least an audiovisual datum, wherein generating the video record further comprises: identifying a plurality of temporal sections of the video file; classifying each temporal section of the plurality of temporal sections to a resume prompt of a plurality of resume prompts, where the plurality of resume prompts is arranged in a prompt ordering; and assembling the plurality of classified temporal sections into the video record using the prompt ordering.

In another aspect a method for automated video record generation includes: capturing, by plurality of sensors, at least audiovisual data from a user; generating, by plurality of sensors, at least an audiovisual datum; converting, by computing device, the at least an audiovisual datum into a video file; generating, by processor, a video record of a user based on the at least an audiovisual datum, wherein assembling the video record further comprises: identifying, by processor, a plurality of temporal sections of the video file; classifying, by processor, each temporal section of the plurality of temporal sections to a resume prompt of a plurality of resume prompts, where the plurality of resume prompts is arranged in a prompt ordering; and assembling, by processor, the plurality of classified temporal sections into the video record using the prompt ordering.

The details of one or more variations of the subject matter described herein are set forth in the accompanying drawings and the description below. Other features and advantages of the subject matter described herein will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

For the purpose of illustrating the invention, the drawings show aspects of one or more embodiments of the invention. However, it should be understood that the present invention is not limited to the precise arrangements and instrumentalities shown in the drawings, wherein:

FIG. 1 is a block diagram of an embodiment of an apparatus for automated video record generation;

FIG. 2 is a block diagram of exemplary machine-learning processes;

FIG. 3 illustrates an exemplary neural network;

FIG. 4 is a block diagram of an exemplary node;

FIG. 5 is a graph illustrating an exemplary relationship between fuzzy sets;

FIG. 6 is a flow diagram of an exemplary method for job forecasting;

FIG. 7 is a block diagram of an embodiment of an apparatus for creating a video record;

FIG. 8 is a block diagram of an exemplary immutable sequential listing;

FIG. 9 is a flow diagram of an exemplary method for creating a video record;

5 FIG. 10 is a block diagram of an embodiment of an apparatus for matching video records with postings;

FIG. 11 is a flow diagram of an exemplary method for matching video records with postings;

FIG. 12 is a block diagram of an exemplary apparatus for parsing and comparing video resume content to target video resume content;

10 FIG. 13 illustrates an exemplary image frame;

FIG. 14 is a table representing an exemplary association between multiple video resumes and a target video resume;

FIG. 15 illustrates an exemplary video resume database using a block diagram;

FIG. 16 illustrates by way of a block diagram an exemplary training data database;

15 FIG. 17 is a flow diagram of an exemplary method of parsing and comparing solicitation video resume content to target video resume;

FIG. 1 is a block diagram of an embodiment of an apparatus for job forecasting;

FIG. 8 is a flow diagram of an exemplary method for job forecasting;

20 FIG. 1 is a block diagram of an embodiment of an apparatus for linking posting data to a plurality of user identifiers;

FIG. 6 is a flow diagram of an exemplary method for linking posting data to a plurality of user identifiers;

25 FIG. 7 is a block diagram of a computing system that can be used to implement any one or more of the methodologies disclosed herein and any one or more portions thereof. The drawings are not necessarily to scale and may be illustrated by phantom lines, diagrammatic representations, and fragmentary views. In certain instances, details that are not necessary for an understanding of the embodiments or that render other details difficult to perceive may have been omitted. Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

At a high level, aspects of the present disclosure are directed to apparatuses and methods for automated video record generation. In an embodiment, video records may represent a user more wholistically than a tradition resume. Video records may allow a user to show a more
5 dynamic version of themselves than a traditional paper resume. Aspects of the present disclosure can be used to generate prompts for users to answer. Aspects of the present disclosure can also be used to receive audiovisual data from the user. This is so, at least in part, because of the plurality of sensors. Exemplary embodiments illustrating aspects of the present disclosure are described below in the context of several specific examples.

Referring now to FIG. 1, an exemplary embodiment of an apparatus 100 for
10 automatically generating video records. Apparatus 100 includes a plurality of sensors. As used in this disclosure, a “sensor” is a device that is configured to detect a phenomenon and transmit information and/or datum related to the detection of the phenomenon. For instance, and without limitation, a sensor 136 may transform an electrical and/or nonelectrical stimulation into an
15 electrical signal that is suitable to be processed by an electrical circuit, such as a processor 104. A sensor 136 may generate a sensor output signal, which transmits information and/or datum related to a detection by the sensor. A sensor output signal may include any signal form described in this disclosure, such as for example, digital, analog, optical, electrical, fluidic, and the like. In some cases, a sensor, a circuit, and/or a controller may perform one or more signal
20 processing steps on a signal. For instance, a sensor, circuit, and/or controller may analyze, modify, and/or synthesize a signal in order to improve the signal, for instance by improving transmission, storage efficiency, or signal to noise ratio. Plurality of sensors includes an auditory sensor and an optical sensor. In an embodiment, an auditory sensor may include microphones, and other sensors that detect sound. In an embodiment, an optical sensor may include sensors
25 that convert light rays into electrical signals such as cameras, infrared sensors, photodetectors, motion sensors, and the like. The plurality of sensors may be physically isolated from each other. “Physical isolation,” for the purposes of this disclosure, refers to a first system’s components, communicative connection, and/or any other constituent parts, whether software or hardware, are separated from a second system’s components, communicative connection, and any other
30 constituent parts, whether software or hardware, respectively. The plurality of sensors is

configured to detect at least audiovisual data 112 from a user 108. The plurality of sensors may be attached to the processor 104, discussed in further detail below, or they may be a separate entity and communicatively connected to a processor 104. A user 108 may have access to the plurality of sensors such that the sensors may detect at least audiovisual data 112 from a user 108. As used in this disclosure, “audiovisual” relates to information that includes images, verbal content, non-verbal content, audio content, as well as any other information gathered from auditory and optical sensors. In an embodiment, audiovisual data 112 may include user movements, sounds, voices, videos, written data, or any other data that may occur to persons skilled in the art upon reviewing the entirety of this disclosure.

Still referring to FIG. 1, each sensor 136 in the plurality of sensors is configured to generate at least an audiovisual datum 124 based on the at least audiovisual data 112 from a user 108. In one or more embodiments, each sensor 136 of the plurality of sensors may generate a sensor output signal that includes information and/or datum related to the detected event and/or phenomenon, such as, for example, audiovisual datum 124 of the plurality of sensors respectively. For the purposes of this disclosure, “audiovisual datum” is an electrical signal representation information and/or datum of a detected event associated with the audiovisual data 112. For example, and without limitation, audiovisual datum 124 may include audio recording of user speak, video recording of user, and the like.

Still referring to FIG. 1, apparatus 100 includes a processor 104 communicatively connected to the plurality of sensors. As used herein, “communicatively connected” is a process whereby one device, component, or circuit is able to receive data from and/or transmit data to another device, component, or circuit. In an embodiment, communicative connecting includes electrically connecting at least an output of one device, component, or circuit to at least an input of another device, component, or circuit. For example, and without limitation, each sensor 136 of the plurality of sensors may be communicatively connected to processor 104. In one or more embodiments, a communicative connection between processor 104 and sensor 136 may be wireless and/or wired. For example, and without limitation, processor 104 and sensor 136 may communicative via a controller area network (CAN) communication. processor 104 may include any computing device as described in this disclosure, including without limitation a microcontroller, microprocessor, digital signal processor (DSP) and/or system on a chip (SoC) as

described in this disclosure. Computing device may include, be included in, and/or communicate with a mobile device such as a mobile telephone or smartphone. Processor 104 may include a single computing device operating independently, or may include two or more computing device operating in concert, in parallel, sequentially or the like; two or more computing devices may be included together in a single computing device or in two or more computing devices. Processor 104 may interface or communicate with one or more additional devices as described below in further detail via a network interface device. Network interface device may be utilized for connecting processor 104 to one or more of a variety of networks, and one or more devices. Examples of a network interface device include, but are not limited to, a network interface card (e.g., a mobile network interface card, a LAN card), a modem, and any combination thereof. Examples of a network include, but are not limited to, a wide area network (e.g., the Internet, an enterprise network), a local area network (e.g., a network associated with an office, a building, a campus or other relatively small geographic space), a telephone network, a data network associated with a telephone/voice provider (e.g., a mobile communications provider data and/or voice network), a direct connection between two computing devices, and any combinations thereof. A network may employ a wired and/or a wireless mode of communication. In general, any network topology may be used. Information (e.g., data, software etc.) may be communicated to and/or from a computer and/or a computing device. Processor 104 may include but is not limited to, for example, a computing device or cluster of computing devices in a first location and a second computing device or cluster of computing devices in a second location. Processor 104 may include one or more computing devices dedicated to data storage, security, distribution of traffic for load balancing, and the like. Processor 104 may distribute one or more computing tasks as described below across a plurality of computing devices of computing device, which may operate in parallel, in series, redundantly, or in any other manner used for distribution of tasks or memory between computing devices. Processor 104 may be implemented using a “shared nothing” architecture in which data is cached at the worker, in an embodiment, this may enable scalability of apparatus 100 and/or computing device.

With continued reference to FIG. 1, processor 104 may be designed and/or configured to perform any method, method step, or sequence of method steps in any embodiment described in this disclosure, in any order and with any degree of repetition. For instance, processor 104 may

be configured to perform a single step or sequence repeatedly until a desired or commanded outcome is achieved; repetition of a step or a sequence of steps may be performed iteratively and/or recursively using outputs of previous repetitions as inputs to subsequent repetitions, aggregating inputs and/or outputs of repetitions to produce an aggregate result, reduction or
5 decrement of one or more variables such as global variables, and/or division of a larger processing task into a set of iteratively addressed smaller processing tasks. Processor 104 may perform any step or sequence of steps as described in this disclosure in parallel, such as simultaneously and/or substantially simultaneously performing a step two or more times using two or more parallel threads, processor cores, or the like; division of tasks between parallel
10 threads and/or processes may be performed according to any protocol suitable for division of tasks between iterations. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which steps, sequences of steps, processing tasks, and/or data may be subdivided, shared, or otherwise dealt with using iteration, recursion, and/or parallel processing.

Continuing to refer to FIG. 1, processor 104 may be configured to receive an employment datum 120. For the purpose of this disclosure, “employment datum” is information related to an available and/or open job position. For the purposes of this disclosure, a “job position” (also referred to in this disclosure as a “job”) is a paid occupation with designated tasks associated therewith. A job position may include an employment with an employer, such as work as an
20 employee (part-time or full-time), worker, contractor, self-employed, and the like. For example, and without limitation, employment datum 120 may include information and/or data from a job posting and/or listing that describes an open job position. Employment datum may include a job position title, qualifications and/or requirements for the job position, expected responsibilities associated with the job position, benefits with the job position, compensation, geographical
25 location, employer information, and the like. Employment datum may include information related to an employer’s expectations of a person hired for such a job position. For instance, and without limitations, employment datum 120 may include minimum qualifications that a candidate must possess to adequately perform the job position. Qualifications for job position may include education, certification, experience, desired skills and/or abilities, personal qualities,
30 and the like. Employment datum may also include information that a person hired for the job

position may expect from the job position. For instance, and without limitation, employment datum 120 may include working hours for the job position, a type of salary, degree of professionalism, and the like. Employment datum includes previous work history. In an embodiment, previous work history may include job position title, qualifications and/or requirements for the job position, expected responsibilities associated with the job position, benefits with the job position, compensation, geographical location, employer information, and the like related to a previous job a user 108 may have had. In one or more embodiments, employment datum 120 may include a datum or a plurality of data related to an available job. In one or more embodiments, employment datum 120 may be provided to or received by processor 104 using various means. In one or more embodiments, employment datum 120 may be provided to processor 104 by a user 108, such as a job seeker or potential job candidate that is interested in being a candidate or considered for a job position by the employer of the job position. A user 108 may manually input employment datum 120 into processor 104 using, for example, a graphic user interface and/or an input device. For example, and without limitation, a user 108 may use a peripheral input device to navigate graphical user interface and provide employment datum 120 to processor 104. Non-limiting exemplary input devices include keyboards, joy sticks, light pens, tracker balls, scanners, tablet, microphones, mouses, switches, buttons, sliders, touchscreens, and the like. In other embodiments, employment datum 120 may be provided to processor 104 by a database over a network from, for example, a network-based platform. Employment datum may be stored in a database and communicated to processor 104 upon a retrieval request from a user 108 and/or from processor 104. In other embodiments, employment datum 120 may be communicated from a third-party application, such as from a third-party application on a third party server, using a network. For example, employment datum 120 may be downloaded from a hosting website for job listings. In one or more embodiments, processor 104 may extract employment datum 120 from an accumulation of information provided by a database. For instance, and without limitation, processor 104 may extract needed information from database regarding the job position and avoid taking any information determined to be unnecessary. This may be performed by processor 104 using a machine-learning model, which is described in this disclosure further below.

With continued reference to FIG. 1, processor 104 may be configured to generate a prompt 116 based on the employment datum 120. As used in this disclosure, a “prompt” is a brief passage of audiovisual information such as text, images, spoken words or the like, that provides a guide and/or starting point for a user response. In an embodiment, a prompt 116 may include a question pertaining to work history of a user 108, user work skills, user work preferences, and the like. In an embodiment, a prompt 116 may include questions that determine how a user 108 behaves. For example, a prompt 116 may be “discuss a time that you (the user 108) handled conflict in the workplace”. Prompt may be presented as text on a graphical user interface (GUI), discussed in further detail below. In an embodiment, prompt 116 may also be presented as audio, or a combination of visual and audio forms. In another embodiment, prompt 116 may be presented by a virtual avatar. For example, a virtual avatar may be presented as a robotic avatar, a human avatar, a 2D model and/or figure, a 3D model and/or figure, and the like. Computing device is further configured to generate prompt 116 using a virtual assistant. As used herein, a “virtual assistant” is a computer application program that understands natural language voice commands and emulates human interaction. In an embodiment, a virtual assistant may include Siri, Alexa, Google Assistant, Cortana, Bixby, and the like.

Still referring to FIG 1, processor 104 may be further configured to use a machine learning module 200 to generate a prompt 116. A machine learning module 200 may use a machine learning process. A machine learning process, also referred to as a machine-learning algorithm, is a process that automatedly uses training data and/or a training set as described below to generate an algorithm that will be performed by a processor 104 and/or module to produce outputs given data provided as inputs; this is in contrast to a non-machine learning software program where the commands to be executed are determined in advance by a user 108 and written in a programming language. Machine learning module is described in further detail in FIG. 2. Machine learning process may be trained using training data, described in further detail in FIG. 2, to input employment datum 120 and output prompts for a user 108 pertaining to the employment datum 120. In an embodiment, machine learning process may generate a prompt 116 of “what is the hardest dish you have made?” for a user 108 that has a work history of working as a chef. In another embodiment, machine learning process may generate a prompt 116 pertaining to a job position that a user 108 may be applying to. For example, prompt 116 may

ask user 108 to discuss geometric dimensioning and tolerancing for a manufacturing engineering position. Computing device may generate classifier using a classification algorithm, defined as a processes whereby a processor 104 derives, from training data, a model known as a “classifier” for sorting inputs into categories or bins of data. Classification may be performed using, without
5 limitation, linear classifiers such as without limitation logistic regression and/or naive Bayes classifiers, nearest neighbor classifiers such as k-nearest neighbors classifiers, support vector machines, least squares support vector machines, fisher’s linear discriminant, quadratic classifiers, decision trees, boosted trees, random forest classifiers, learning vector quantization, and/or neural network-based classifiers. Classifier may categorize employment datum 120 of a
10 user 108 and/or prompts such as experience, work skills, people skills, problem solving, managements skills, and the like. In an embodiment, processor 104 may use a machine learning module 200 to classify employment datum 120 to a template comprising of a list of prompts in a specific order. Template may be generated by a machine learning module 200 in ways discussed above. Additionally, template may be generated by a human and stored in processor 104 for
15 machine learning module 200 to use. Template may comprise of a specific order to present prompts to user 108 such that generated a video record (discussed below) may utilize the specific order to assemble video clips.

Still referring to FIG. 1, processor 104 may use a language processing model to extract words from the employment datum 120. Language processing module may include any hardware
20 and/or software module. Language processing module may be configured to extract, from the one or more documents, one or more words. One or more words may include, without limitation, strings of one or more characters, including without limitation any sequence or sequences of letters, numbers, punctuation, diacritic marks, engineering symbols, geometric dimensioning and tolerancing (GD&T) symbols, chemical symbols and formulas, spaces, whitespace, and other
25 symbols, including any symbols usable as textual data as described above. Textual data may be parsed into tokens, which may include a simple word (sequence of letters separated by whitespace) or more generally a sequence of characters as described previously. The term “token,” as used herein, refers to any smaller, individual groupings of text from a larger source of text; tokens may be broken up by word, pair of words, sentence, or other delimitation. These
30 tokens may in turn be parsed in various ways. Textual data may be parsed into words or

sequences of words, which may be considered words as well. Textual data may be parsed into "n-grams", where all sequences of n consecutive characters are considered. Any or all possible sequences of tokens or words may be stored as "chains", for example for use as a Markov chain or Hidden Markov Model.

5 Still referring to FIG. 1, language processing module may operate to produce a language processing model. Language processing model may include a program automatically generated by processor 104 and/or language processing module to produce associations between one or more words extracted from at least a document and detect associations, including without limitation mathematical associations, between such words. Associations between language
10 elements, where language elements include for purposes herein extracted words, relationships of such categories to other such term may include, without limitation, mathematical associations, including without limitation statistical correlations between any language element and any other language element and/or language elements. Statistical correlations and/or mathematical associations may include probabilistic formulas or relationships indicating, for instance, a
15 likelihood that a given extracted word indicates a given category of semantic meaning. As a further example, statistical correlations and/or mathematical associations may include probabilistic formulas or relationships indicating a positive and/or negative association between at least an extracted word and/or a given semantic meaning; positive or negative indication may include an indication that a given document is or is not indicating a category semantic meaning.
20 Whether a phrase, sentence, word, or other textual element in a document or corpus of documents constitutes a positive or negative indicator may be determined, in an embodiment, by mathematical associations between detected words, comparisons to phrases and/or words indicating positive and/or negative indicators that are stored in memory at processor 104, or the like.

25 Still referring to FIG. 1, language processing module and/or diagnostic engine may generate the language processing model by any suitable method, including without limitation a natural language processing classification algorithm; language processing model may include a natural language process classification model that enumerates and/or derives statistical relationships between input terms and output terms. Algorithm to generate language processing
30 model may include a stochastic gradient descent algorithm, which may include a method that

iteratively optimizes an objective function, such as an objective function representing a statistical estimation of relationships between terms, including relationships between input terms and output terms, in the form of a sum of relationships to be estimated. In an alternative or additional approach, sequential tokens may be modeled as chains, serving as the observations in a Hidden Markov Model (HMM). HMMs as used herein are statistical models with inference algorithms that that may be applied to the models. In such models, a hidden state to be estimated may include an association between an extracted words, phrases, and/or other semantic units. There may be a finite number of categories to which an extracted word may pertain; an HMM inference algorithm, such as the forward-backward algorithm or the Viterbi algorithm, may be used to estimate the most likely discrete state given a word or sequence of words. Language processing module may combine two or more approaches. For instance, and without limitation, machine-learning program may use a combination of Naive-Bayes (NB), Stochastic Gradient Descent (SGD), and parameter grid-searching classification techniques; the result may include a classification algorithm that returns ranked associations.

Continuing to refer to FIG. 1, generating language processing model may include generating a vector space, which may be a collection of vectors, defined as a set of mathematical objects that can be added together under an operation of addition following properties of associativity, commutativity, existence of an identity element, and existence of an inverse element for each vector, and can be multiplied by scalar values under an operation of scalar multiplication compatible with field multiplication, and that has an identity element is distributive with respect to vector addition, and is distributive with respect to field addition. Each vector in an n-dimensional vector space may be represented by an n-tuple of numerical values. Each unique extracted word and/or language element as described above may be represented by a vector of the vector space. In an embodiment, each unique extracted and/or other language element may be represented by a dimension of vector space; as a non-limiting example, each element of a vector may include a number representing an enumeration of co-occurrences of the word and/or language element represented by the vector with another word and/or language element. Vectors may be normalized, scaled according to relative frequencies of appearance and/or file sizes. In an embodiment associating language elements to one another as described above may include computing a degree of vector similarity between a vector representing each

language element and a vector representing another language element; vector similarity may be measured according to any norm for proximity and/or similarity of two vectors, including without limitation cosine similarity, which measures the similarity of two vectors by evaluating the cosine of the angle between the vectors, which can be computed using a dot product of the two vectors divided by the lengths of the two vectors. Degree of similarity may include any other geometric measure of distance between vectors.

Still referring to FIG. 1, language processing module may use a corpus of documents to generate associations between language elements in a language processing module, and diagnostic engine may then use such associations to analyze words extracted from one or more documents and determine that the one or more documents indicate significance of a category. In an embodiment, language module and/or processor 104 may perform this analysis using a selected set of significant documents, such as documents identified by one or more experts as representing good information; experts may identify or enter such documents via graphical user interface, or may communicate identities of significant documents according to any other suitable method of electronic communication, or by providing such identity to other persons who may enter such identifications into processor 104. Documents may be entered into a computing device by being uploaded by an expert or other persons using, without limitation, file transfer protocol (FTP) or other suitable methods for transmission and/or upload of documents; alternatively or additionally, where a document is identified by a citation, a uniform resource identifier (URI), uniform resource locator (URL) or other datum permitting unambiguous identification of the document, diagnostic engine may automatically obtain the document using such an identifier, for instance by submitting a request to a database or compendium of documents such as JSTOR as provided by Ithaka Harbors, Inc. of New York.

Continuing to reference FIG. 1, processor 104 may be configured to receive the at least an audiovisual datum and convert the at least an audiovisual datum into a video file. Computing device may receive at least an audiovisual datum from a sensor 136 or plurality of sensors after presenting the prompt 116 to the user 108. Audiovisual datum may be stored in a database such that the processor 104 may extract needed information at any time. Processor 104 is configured to generate a video record 128 of the user 108 based on the at least an audiovisual datum. As used in this disclosure, a “video record” is an item of digital media in visual and/or audio form to

provide a recording promoting a subject. For example, a video record may include a video resume. As used in this disclosure, a “subject” is a person, for example a jobseeker. In some cases, video record 128 may include content that is representative or communicative of at least an attribute of the subject. Attributes may include the subject’s skills, competencies, credentials, talents, and the like. In some cases, attributes may be explicitly conveyed within video record 128. Alternatively, or additionally, in some cases, attributes may be conveyed implicitly within video record 128. The subject may be represented directly by video record 128. For example, in some cases, an image component may represent a visual of the subject. As used in this disclosure, an “image component” may be a visual representation of information, such as a plurality of temporally sequential frames and/or pictures, related to a video record. For example, an image component may include animations, still imagery, recorded video, and the like. In some cases, video record 128 may include a digital video, which may be communicated by way of digital signals, for example between computing devices which are communicatively connected with at least a network. To optimize speed and/or cost of transmission of video record 128. Video may be compressed according to a video compression coding format (i.e., codec). Exemplary video compression codecs include H.26x codecs, MPEG formats, VVC, SVT-AV1, and the like. In some cases, compression of a digital video may be lossy, in which some information may be lost during compression. Alternatively, or additionally, in some cases, compression of a digital video may be substantially lossless, where substantially no information is lost during compression.

Continuing to refer to FIG. 1, generating the video record 128 further includes identifying a plurality of temporal sections of the video file created from the audiovisual datum. As used herein, a “temporal section” is a clip of a video file that is marked by a start and end time of the whole video file. A plurality of temporal sections may be identified using a neural network, discussed in further detail in FIG. 3. A neural network may be trained to output temporal sections of the video file. A temporal section may be user defined such that a user may input into a processor, temporal sections of a video file. A temporal section may be defined in any other way is contemplated within the scope of this disclosure. Temporal sections may be based on resume prompt 116 (also referred to as “prompt”) such that each section has a clip of the user answering a prompt 116. Neural network may be training by inputting training examples of

videos partitioned by hand, wherein the start of the temporal section is the prompt 116 and the end of the temporal section is the end of the user's answer to the prompt 116. Neural network may be trained to recognize the start of a temporal section by the presentation of a title card of the prompt and the end of a temporal section as the start of the next title card. As used herein, a
5 "title card" is an audiovisual representation of a prompt. In an embodiment, a title card may have the prompt written on a colored background before showing a user 108 answering the prompt.

Continuing to refer to FIG. 1, generating the video record 128 further includes classifying each temporal section of the plurality of temporal sections to a resume prompt of a plurality of resume prompts, where the plurality of resume prompts is arranged in a prompt ordering.

10 Temporal sections extracted from the audiovisual datum video file may be classified to resume prompts. In some cases, processor 104 may use audiovisual speech recognition processes (AVSR) to recognize verbal content in temporal sections of the video file. AVSR is discussed in further detail below. Processor 104 may use AVSR to recognize a presented resume prompt 116 and associate prompt 116 with the temporal section. In other cases, processor 104 may use
15 optical character recognition or optical character reader (OCR) including automatic conversion of images of written (e.g., typed, handwritten or printed text) into machine-encoded text. In this case, processor 104 may recognize a title card and associate the title card with a prompt 116.

Processor 104 may use a classification algorithm to classify temporal sections into bins, wherein the prompt 116 is the classifier and the bins of data contain the temporal sections related to the
20 prompt 116. Additionally, the plurality of resume prompts and the associated temporal sections are arranged in a prompt ordering. In an embodiment, processor 104 may take the ordering of a template of prompts (discussed in an example above) and organize temporal sections into the ordering of the template. Template of prompts may be used by a classification algorithm to determine the prompts to present to a user 108. In some cases, processor 104 may organize
25 temporal sections from general prompts into more job specific prompts. For example, processor 104 may organize temporal sections starting from prompts about a user's background to more technical questions, like questions about job related tasks. Prompt ordering may be user determined such that the user 108 may have the option to view temporal sections and determine the order that is presented in video record 128. Processor 104 assembles the plurality of

classified temporal sections into the video record 128 using the prompt ordering that may be user or machine determined.

Still referring to FIG. 1, temporal sections may be compressed to be assembled into a video record 128. In some embodiments, comparing series of frames may include video
5 compression by inter-frame coding. The “inter” part of the term refers to the use of inter frame prediction. This kind of prediction tries to take advantage from temporal redundancy between neighboring frames enabling higher compression rates. Video data compression is the process of encoding information using fewer bits than the original representation. Any compression is either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical
10 redundancy. No information is lost in lossless compression. Lossy compression reduces bits by removing unnecessary or less important information. Typically, a device that performs data compression is referred to as an encoder, and one that performs the reversal of the process (decompression) as a decoder. Compression is useful because it reduces the resources required to store and transmit data. Computational resources are consumed in the compression and
15 decompression processes. Data compression is subject to a space–time complexity trade-off. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in full before watching it may be inconvenient or require additional storage. Video data may be represented as a series of still image frames. Such data usually
20 contains abundant amounts of spatial and temporal redundancy. Video compression algorithms attempt to reduce redundancy and store information more compactly.

Still referring to FIG. 1, inter-frame coding works by comparing each frame in the video with the previous one. Individual frames of a video sequence are compared from one frame to the next, and the video compression codec sends only the differences to the reference frame. If
25 the frame contains areas where nothing has moved, the system can simply issue a short command that copies that part of the previous frame into the next one. If sections of the frame move in a simple manner, the compressor can emit a (slightly longer) command that tells the decompressor to shift, rotate, lighten, or darken the copy. Usually, the encoder will also transmit a residue signal which describes the remaining more subtle differences to the reference imagery. Using
30 entropy coding, these residue signals have a more compact representation than the full signal. In

areas of video with more motion, the compression must encode more data to keep up with the larger number of pixels that are changing. As used in this disclosure, reference frames are frames of a compressed video (a complete picture) that are used to define future frames. As such, they are only used in inter-frame compression techniques. Some modern video encoding standards, such as H.264/AVC, allow the use of multiple reference frames. This allows the video encoder to choose among more than one previously decoded frame on which to base each macroblock in the next frame. While the best frame for this purpose is usually the previous frame, the extra reference frames can improve compression efficiency and/or video quality. The two frame types used in inter-frame coding is P-frames and B-frames. A P-frame (Predicted picture) holds only the changes in the image from the previous frame. For example, in a scene where a car moves across a stationary background, only the car's movements need to be encoded. The encoder does not need to store the unchanging background pixels in the P-frame, thus saving space. A B-frame (Bidirectional predicted picture) saves even more space by using differences between the current frame and both the preceding and following frames to specify its content. An inter coded frame is divided into blocks known as macroblocks. A macroblock is a processing unit in image and video compression formats based on linear block transforms, typically the discrete cosine transform (DCT). A macroblock typically consists of 16×16 samples, and is further subdivided into transform blocks, and may be further subdivided into prediction blocks. Formats which are based on macroblocks include JPEG, where they are called MCU blocks, H.261, MPEG-1 Part 2, H.262/MPEG-2 Part 2, H.263, MPEG-4 Part 2, and H.264/MPEG-4 AVC. After the inter coded frame is divided into macroblocks, instead of directly encoding the raw pixel values for each block, the encoder will try to find a block similar to the one it is encoding on a previously encoded frame, referred to as a reference frame. This process is done by a block matching algorithm. If the encoder succeeds on its search, the block could be encoded by a vector, known as motion vector, which points to the position of the matching block at the reference frame. The process of motion vector determination is called motion estimation. In most cases the encoder will succeed, but the block found is likely not an exact match to the block it is encoding. This is why the encoder will compute the differences between them. Those residual values are known as the prediction error and need to be transformed and sent to the decoder. To sum up, if the encoder succeeds in finding a matching block on a reference frame, it will obtain a motion vector

pointing to the matched block and a prediction error. Using both elements, the decoder will be able to recover the raw pixels of the block. For example, audiovisual datum video file may be compressed using a P-frame algorithm and broken down into macroblocks. Individual still images taken from video record 128 can then be compared against a reference frame taken from the video file. A P-frame from video record 128 would only hold the changes in image from video file. Exemplary video compression codecs include without limitation H.26x codecs, MPEG formats, VVC, SVT-AV1, and the like. In some cases, compression may be lossy, in which some information may be lost during compression. Alternatively, or additionally, in some cases, compression may be substantially lossless, where substantially no information is lost during compression. In some cases, image component 116a-b may include a plurality of temporally sequential frames. In some cases, each frame may be encoded (e.g., bitmap or vector-based encoding). Each frame may be configured to be displayed by way of a display. Exemplary displays include without limitation light emitting diode (LED) displays, cathode ray tube (CRT) displays, liquid crystal displays (LCDs), organic LEDs (OLEDs), quantum dot displays, projectors (e.g., scanned light projectors), and the like.

Still referring to FIG. 1, video record 128 may be representative subject-specific data. As used in this disclosure, “subject-specific data” is any element of information that is associated with a specific subject. Exemplary forms of subject-specific data include image component, video record 128, non-verbal content, verbal content, audio component, as well as any information derived directly or indirectly from video record 128 or any other subject-specific data. For example, subject-specific data could be the physical properties of subject, such as their body posture or facial expression. Subject-specific data could also be audio sensory properties of subject, such as tone of voice or background audio in a resume video.

In some cases, video record 128 may include non-verbal content. As used in this disclosure, “non-verbal content” is all communication that is not characterized as verbal content. As used in this disclosure, “verbal content” is comprehensible language-based communication. For example, verbal content may include “visual verbal content” which is literal and/or written verbal content. Non-verbal content 132 includes all forms of communication which are not conveyed with use of language. Exemplary non-verbal content may include change in intonation and/or stress in a speaker’s voice, expression of emotion, and the like. For example, in some

cases, non-verbal content may include visual non-verbal content. As used in this disclosure, “visual non-verbal content” is non-verbal content that is visually represented. In some cases, visual non-verbal content may be included within video record 128 by way of image component.

In some cases, a non-verbal classifier may classify non-verbal content present in one or more image component to one or more of video record 128, a feature. Non-verbal classifier may be used with a machine learning algorithm to create a video record 128. Non-verbal classifier may include a number of classifiers, for example each being tasked with classifying a particular attribute or form of non-verbal content. For example, in some cases, non-verbal classifier may classify a video record 128 and related subject as associated with a feature representative of ‘personable.’ Non-verbal classifier may include another specialized visual non-verbal classifier to classify visual non-verbal content as appearing ‘personable’ that is, for example, as having appropriate posture, facial expressions, manner of dress, and the like. In some cases, classifier may include or a constituent part of tree structure, for making associations based upon video record 128.

With continued reference to FIG. 1, in some embodiments, image component -b may include or otherwise represent verbal content. For instance, written or visual verbal content may be included within image component. Visual verbal content may include images of written text represented by image component. For example, visual verbal content may include, without limitation, digitally generated graphics, images of written text (e.g., typewritten, and the like), signage, and the like.

Still referring to FIG. 1, in some embodiments, image component may include or otherwise represent audible verbal content related to at least an attribute of user 108. As used in this disclosure, “audible verbal content” is oral (e.g., spoken) verbal content. In some cases, audible verbal content may be included within video record 128 by way of an audio component. As used in this disclosure, an “audio component” is a representation of audio, for example a sound, a speech, and the like. In some cases, verbal content may be related to at least an attribute of subject. Additionally, or alternatively, visual verbal content and audible verbal content may be used as inputs to classifiers as described throughout this disclosure.

Continuing to reference FIG. 1, video record 128 includes a written component. In an embodiment, a written component may include a PDF or a word version of the video record 128.

In an embodiment, a written component may be the transcription of the video record 128. Alternatively or additionally, a written component may include supplementary information submitted by the user 108. Written component may include work history, work experience, credentials, work objectives, work expectations, and the like. Computing device is configured to

5 transcribe video record 128. In some cases, processor 104 may include audiovisual speech recognition (AVSR) processes to recognize verbal content in video records. For example, processor 104 may use image content to aid in recognition of audible verbal content such as viewing subject move their lips to speak on video to process the audio content of video record 128. AVSR may use image component to aid the overall translation of the audio verbal content

10 of video records. In some embodiments, AVSR may include techniques employing image processing capabilities in lip reading to aid speech recognition processes. In some cases, AVSR may be used to decode (i.e., recognize) indeterministic phonemes or help in forming a preponderance among probabilistic candidates. In some cases, AVSR may include an audio-based automatic speech recognition process and an image-based automatic speech recognition

15 process. AVSR may combine results from both processes with feature fusion. Audio-based speech recognition process may analysis audio according to any method described herein, for instance using a Mel frequency cepstrum coefficients (MFCCs) and/or log-Mel spectrogram derived from raw audio samples. Image-based speech recognition may perform feature recognition to yield an image vector. In some cases, feature recognition may include any feature

20 recognition process described in this disclosure, for example a variant of a convolutional neural network. In some cases, AVSR employs both an audio datum and an image datum to recognize verbal content 136. For instance, audio vector and image vector may each be concatenated and used to predict speech made by a user 108, who is ‘on camera.’

In some cases, processor 104 may be configured to recognize at least a keyword as a

25 function of visual verbal content. In some cases, recognizing at least keyword may include optical character recognition. As used in this disclosure, a “keyword” is an element of word or syntax used to identify and/or match elements to each other. In some cases, processor 104 may transcribe much or even substantially all verbal content from resume video. Video records transcription is illustrated further below in this disclosure.

Still referring to FIG. 1, processor 104 includes a display component 132. In one or more embodiments, display component 132 may be communicatively connected to processor 104. Display component 132 may be remote to processor 104 or integrated into processor 104. Communication between processor 104 and display component 132 may be wired or wireless. In one or more embodiments, display component 132 may be configured to display employment datum 120, audiovisual datum, database information, video record 128, and the like. Display component 132 may include a graphic user interface (GUI) that a user 108 may use to navigate through presented data or information by processor 104. GUI is configured to display the video record 128. In one or more embodiments, a GUI may include a plurality of lines, images, symbols, and the like to show information and/or data. In non-limiting embodiments, display component 132 may include a smartphone, tablet, laptop, desktop, monitor, tablet, touchscreen, head-up display (HUD), and the like. In one or more embodiments, display component 132 may include a screen such as a liquid crystal display (LCD) various other types of displays or monitors, as previously mentioned in this disclosure. In one or more embodiments, user 108 may view information and/or data displayed on display component 132 in real time. In one or more embodiments, display component 132 may be configured to display received or determined information, which may be toggled through using, for example, an input device of display component 132 or processor 104. Display component 132 may include electronic components utilized to display image data or information, such as a video, GUI, photo, and the like.

Referring now to FIG. 2, an exemplary embodiment of a machine-learning module 200 that may perform one or more machine-learning processes as described in this disclosure is illustrated. Machine-learning module may perform determinations, classification, and/or analysis steps, methods, processes, or the like as described in this disclosure using machine learning processes. A “machine learning process,” as used in this disclosure, is a process that automatedly uses training data 204 to generate an algorithm that will be performed by a processor 104/module to produce outputs 208 given data provided as inputs 212; this is in contrast to a non-machine learning software program where the commands to be executed are determined in advance by a user 108 and written in a programming language.

Still referring to FIG. 2, “training data,” as used herein, is data containing correlations that a machine-learning process may use to model relationships between two or more categories

of data elements. For instance, and without limitation, training data 204 may include a plurality of data entries, each entry representing a set of data elements that were recorded, received, and/or generated together; data elements may be correlated by shared existence in a given data entry, by proximity in a given data entry, or the like. Multiple data entries in training data 204 may evince one or more trends in correlations between categories of data elements; for instance, and without limitation, a higher value of a first data element belonging to a first category of data element may tend to correlate to a higher value of a second data element belonging to a second category of data element, indicating a possible proportional or other mathematical relationship linking values belonging to the two categories. Multiple categories of data elements may be related in training data 204 according to various correlations; correlations may indicate causative and/or predictive links between categories of data elements, which may be modeled as relationships such as mathematical relationships by machine-learning processes as described in further detail below. Training data 204 may be formatted and/or organized by categories of data elements, for instance by associating data elements with one or more descriptors corresponding to categories of data elements. As a non-limiting example, training data 204 may include data entered in standardized forms by persons or processes, such that entry of a given data element in a given field in a form may be mapped to one or more descriptors of categories. Elements in training data 204 may be linked to descriptors of categories by tags, tokens, or other data elements; for instance, and without limitation, training data 204 may be provided in fixed-length formats, formats linking positions of data to categories such as comma-separated value (CSV) formats and/or self-describing formats such as extensible markup language (XML), JavaScript Object Notation (JSON), or the like, enabling processes or devices to detect categories of data.

Alternatively or additionally, and continuing to refer to FIG. 2, training data 204 may include one or more elements that are not categorized; that is, training data 204 may not be formatted or contain descriptors for some elements of data. Machine-learning algorithms and/or other processes may sort training data 204 according to one or more categorizations using, for instance, natural language processing algorithms, tokenization, detection of correlated values in raw data and the like; categories may be generated using correlation and/or other processing algorithms. As a non-limiting example, in a corpus of text, phrases making up a number “n” of compound words, such as nouns modified by other nouns, may be identified according to a

statistically significant prevalence of n-grams containing such words in a particular order; such an n-gram may be categorized as an element of language such as a “word” to be tracked similarly to single words, generating a new category as a result of statistical analysis. Similarly, in a data entry including some textual data, a person’s name may be identified by reference to a list, dictionary, or other compendium of terms, permitting ad-hoc categorization by machine-learning algorithms, and/or automated association of data in the data entry with descriptors or into a given format. The ability to categorize data entries automatically may enable the same training data 204 to be made applicable for two or more distinct machine-learning algorithms as described in further detail below. Training data 204 used by machine learning module 200 may correlate any input data as described in this disclosure to any output data as described in this disclosure.

Further referring to FIG. 2, training data may be filtered, sorted, and/or selected using one or more supervised and/or unsupervised machine-learning processes and/or models as described in further detail below; such models may include without limitation a training data classifier 216.

Training data classifier 216 may include a “classifier,” which as used in this disclosure is a machine-learning model as defined below, such as a mathematical model, neural net, or program generated by a machine learning algorithm known as a “classification algorithm,” as described in further detail below, that sorts inputs into categories or bins of data, outputting the categories or bins of data and/or labels associated therewith. A classifier may be configured to output at least a datum 112 that labels or otherwise identifies a set of data that are clustered together, found to be close under a distance metric as described below, or the like. Machine learning module 200 may generate a classifier using a classification algorithm, defined as a processes whereby a processor 104 and/or any module and/or component operating thereon derives a classifier from training data 204. Classification may be performed using, without limitation, linear classifiers such as without limitation logistic regression and/or naive Bayes classifiers, nearest neighbor classifiers such as k-nearest neighbors classifiers, support vector machines, least squares support vector machines, fisher’s linear discriminant, quadratic classifiers, decision trees, boosted trees, random forest classifiers, learning vector quantization, and/or neural network-based classifiers. As a non-limiting example, training data classifier 216 may classify elements of training data to according

to fields of job description for instance, title, role, organization, requisite experience, requisite credentials, and the like.

Still referring to FIG. 2, machine learning module 200 may be configured to perform a lazy-learning process 220 and/or protocol, which may alternatively be referred to as a “lazy loading” or “call-when-needed” process and/or protocol, may be a process whereby machine learning is conducted upon receipt of an input to be converted to an output, by combining the input and training set to derive the algorithm to be used to produce the output on demand. For instance, an initial set of simulations may be performed to cover an initial heuristic and/or “first guess” at an output and/or relationship. As a non-limiting example, an initial heuristic may include a ranking of associations between inputs and elements of training data 204. Heuristic may include selecting some number of highest-ranking associations and/or training data 204 elements. Lazy learning may implement any suitable lazy learning algorithm, including without limitation a K-nearest neighbors algorithm, a lazy naïve Bayes algorithm, or the like; persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various lazy-learning algorithms that may be applied to generate outputs as described in this disclosure, including without limitation lazy learning applications of machine-learning algorithms as described in further detail below.

Alternatively or additionally, and with continued reference to FIG. 2, machine-learning processes as described in this disclosure may be used to generate machine-learning models 224. A “machine-learning model,” as used in this disclosure, is a mathematical and/or algorithmic representation of a relationship between inputs and outputs, as generated using any machine-learning process including without limitation any process as described above and stored in memory; an input is submitted to a machine-learning model 224 once created, which generates an output based on the relationship that was derived. For instance, and without limitation, a linear regression model, generated using a linear regression algorithm, may compute a linear combination of input data using coefficients derived during machine-learning processes to calculate an output datum. As a further non-limiting example, a machine-learning model 224 may be generated by creating an artificial neural network, such as a convolutional neural network comprising an input layer of nodes, one or more intermediate layers, and an output layer of nodes. Connections between nodes may be created via the process of “training” the network,

in which elements from a training data 204 set are applied to the input nodes, a suitable training algorithm (such as Levenberg-Marquardt, conjugate gradient, simulated annealing, or other algorithms) is then used to adjust the connections and weights between nodes in adjacent layers of the neural network to produce the desired values at the output nodes. This process is sometimes referred to as deep learning.

Still referring to FIG. 2, machine-learning algorithms may include at least a supervised machine-learning process 228. At least a supervised machine-learning process 228, as defined herein, include algorithms that receive a training set relating a number of inputs to a number of outputs, and seek to find one or more mathematical relations relating inputs to outputs, where each of the one or more mathematical relations is optimal according to some criterion specified to the algorithm using some scoring function. For instance, a supervised learning algorithm may include subject-specific data as described above as inputs, description-specific data as outputs, and a scoring function representing a desired form of relationship to be detected between inputs and outputs; scoring function may, for instance, seek to maximize the probability that a given input and/or combination of elements inputs is associated with a given output to minimize the probability that a given input is not associated with a given output. Scoring function may be expressed as a risk function representing an “expected loss” of an algorithm relating inputs to outputs, where loss is computed as an error function representing a degree to which a prediction generated by the relation is incorrect when compared to a given input-output pair provided in training data 204. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various possible variations of at least a supervised machine-learning process 228 that may be used to determine relation between inputs and outputs. Supervised machine-learning processes may include classification algorithms as defined above.

Further referring to FIG. 2, machine learning processes may include at least an unsupervised machine-learning processes 232. An unsupervised machine-learning process, as used herein, is a process that derives inferences in datasets without regard to labels; as a result, an unsupervised machine-learning process may be free to discover any structure, relationship, and/or correlation provided in the data. Unsupervised processes may not require a response variable; unsupervised processes may be used to find interesting patterns and/or inferences

between variables, to determine a degree of correlation between two or more variables, or the like.

Still referring to FIG. 2, machine learning module 200 may be designed and configured to create a machine-learning model 224 using techniques for development of linear regression models. Linear regression models may include ordinary least squares regression, which aims to minimize the square of the difference between predicted outcomes and actual outcomes according to an appropriate norm for measuring such a difference (e.g. a vector-space distance norm); coefficients of the resulting linear equation may be modified to improve minimization. Linear regression models may include ridge regression methods, where the function to be minimized includes the least-squares function plus term multiplying the square of each coefficient by a scalar amount to penalize large coefficients. Linear regression models may include least absolute shrinkage and selection operator (LASSO) models, in which ridge regression is combined with multiplying the least-squares term by a factor of 1 divided by double the number of samples. Linear regression models may include a multi-task lasso model wherein the norm applied in the least-squares term of the lasso model is the Frobenius norm amounting to the square root of the sum of squares of all terms. Linear regression models may include the elastic net model, a multi-task elastic net model, a least angle regression model, a LARS lasso model, an orthogonal matching pursuit model, a Bayesian regression model, a logistic regression model, a stochastic gradient descent model, a perceptron model, a passive aggressive algorithm, a robustness regression model, a Huber regression model, or any other suitable model that may occur to persons skilled in the art upon reviewing the entirety of this disclosure. Linear regression models may be generalized in an embodiment to polynomial regression models, whereby a polynomial equation (e.g., a quadratic, cubic or higher-order equation) providing a best predicted output/actual output fit is sought; similar methods to those described above may be applied to minimize error functions, as will be apparent to persons skilled in the art upon reviewing the entirety of this disclosure.

Continuing to refer to FIG. 2, machine-learning algorithms may include, without limitation, linear discriminant analysis. Machine-learning algorithm may include quadratic discriminate analysis. Machine-learning algorithms may include kernel ridge regression.

Machine-learning algorithms may include support vector machines, including without limitation

support vector classification-based regression processes. Machine-learning algorithms may include stochastic gradient descent algorithms, including classification and regression algorithms based on stochastic gradient descent. Machine-learning algorithms may include nearest neighbors algorithms. Machine-learning algorithms may include various forms of latent space regularization such as variational regularization. Machine-learning algorithms may include Gaussian processes such as Gaussian Process Regression. Machine-learning algorithms may include cross-decomposition algorithms, including partial least squares and/or canonical correlation analysis. Machine-learning algorithms may include naïve Bayes methods. Machine-learning algorithms may include algorithms based on decision trees, such as decision tree classification or regression algorithms. Machine-learning algorithms may include ensemble methods such as bagging meta-estimator, forest of randomized trees, AdaBoost, gradient tree boosting, and/or voting classifier methods. Machine-learning algorithms may include neural network algorithms, including convolutional neural network processes.

Referring now to FIG. 3, an exemplary embodiment of neural network 300 is illustrated. A neural network 300 also known as an artificial neural network, is a network of “nodes,” or data structures having one or more inputs, one or more outputs, and a function determining outputs based on inputs. Such nodes may be organized in a network, such as without limitation a convolutional neural network, including an input layer of nodes, one or more intermediate layers, and an output layer of nodes. Connections between nodes may be created via the process of “training” the network, in which elements from a training dataset are applied to the input nodes, a suitable training algorithm (such as Levenberg-Marquardt, conjugate gradient, simulated annealing, or other algorithms) is then used to adjust the connections and weights between nodes in adjacent layers of the neural network to produce the desired values at the output nodes. This process is sometimes referred to as deep learning. Connections may run solely from input nodes toward output nodes in a “feed-forward” network or may feed outputs of one layer back to inputs of the same or a different layer in a “recurrent network.”

Referring now to FIG. 4, an exemplary embodiment of a node 400 of a neural network is illustrated. Node 400 may include, without limitation a plurality of inputs x_i that may receive numerical values from inputs to a neural network containing the node and/or from other nodes. Node may perform a weighted sum of inputs using weights w_i that are multiplied by respective

inputs x_i . Additionally or alternatively, a bias b may be added to the weighted sum of the inputs such that an offset is added to each unit in the neural network layer that is independent of the input to the layer. The weighted sum may then be input into a function ϕ , which may generate one or more outputs y . Weight w_i applied to an input x_i may indicate whether the input is

5 “excitatory,” indicating that it has strong influence on the one or more outputs y , for instance by the corresponding weight having a large numerical value, and/or a “inhibitory,” indicating it has a weak effect influence on the one more inputs y , for instance by the corresponding weight having a small numerical value. The values of weights w_i may be determined by training a neural network using training data, which may be performed using any suitable process as

10 described above.

Referring to FIG. 5, an exemplary embodiment of fuzzy set comparison 500 is illustrated. A first fuzzy set 504 may be represented, without limitation, according to a first membership function 508 representing a probability that an input falling on a first range of values 512 is a member of the first fuzzy set 504, where the first membership function 508 has values on a range

15 of probabilities such as without limitation the interval $[0,1]$, and an area beneath the first membership function 508 may represent a set of values within first fuzzy set 504. Although first range of values 512 is illustrated for clarity in this exemplary depiction as a range on a single number line or axis, first range of values 512 may be defined on two or more dimensions, representing, for instance, a Cartesian product between a plurality of ranges, curves, axes,

20 spaces, dimensions, or the like. First membership function 508 may include any suitable function mapping first range 512 to a probability interval, including without limitation a triangular function defined by two linear elements such as line segments or planes that intersect at or below the top of the probability interval. As a non-limiting example, triangular membership function may be defined as:

$$25 \quad y(x, a, b, c) = \begin{cases} 0, & \text{for } x > c \text{ and } x < a \\ \frac{x-a}{b-a}, & \text{for } a \leq x < b \\ \frac{c-x}{c-b}, & \text{if } b < x \leq c \end{cases}$$

a trapezoidal membership function may be defined as:

$$y(x, a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right)$$

a sigmoidal function may be defined as:

$$y(x, a, c) = \frac{1}{1 - e^{-a(x-c)}}$$

a Gaussian membership function may be defined as:

$$y(x, c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$$

5 and a bell membership function may be defined as:

$$y(x, a, b, c,) = \left[1 + \left| \frac{x - c}{a} \right|^{2b} \right]^{-1}$$

Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various alternative or additional membership functions that may be used consistently with this disclosure.

10 Still referring to FIG. 5, first fuzzy set 504 may represent any value or combination of values as described above, including output from one or more processes (e.g., machine-learning models), subject-specific data, and description-specific data. A second fuzzy set 516, which may represent any value which may be represented by first fuzzy set 504, may be defined by a second membership function 520 on a second range 524; second range 524 may be identical and/or
 15 overlap with first range 512 and/or may be combined with first range via Cartesian product or the like to generate a mapping permitting evaluation overlap of first fuzzy set 504 and second fuzzy set 516. Where first fuzzy set 504 and second fuzzy set 516 have a region 528 that overlaps, first membership function 508 and second membership function 520 may intersect at a point 532 representing a probability, as defined on probability interval, of a match between first fuzzy set
 20 504 and second fuzzy set 516. Alternatively or additionally, a single value of first and/or second fuzzy set may be located at a locus 536 on first range 512 and/or second range 524, where a probability of membership may be taken by evaluation of first membership function 508 and/or second membership function 520 at that range point. A probability at 528 and/or 532 may be compared to a threshold 540 to determine whether a positive match is indicated. Threshold 540
 25 may, in a non-limiting example, represent a degree of match between first fuzzy set 504 and second fuzzy set 516, and/or single values therein with each other or with either set, which is sufficient for purposes of the matching process; for instance, threshold may indicate a sufficient degree of overlap between an output from one or more machine-learning models and/or subject-

specific data and a predetermined class, such as without limitation a job description, for combination to occur as described above. Alternatively or additionally, each threshold may be tuned by a machine-learning and/or statistical process, for instance and without limitation as described in further detail below.

5 Further referring to FIG. 5, in an embodiment, a degree of match between fuzzy sets may be used to classify a subject 120 with at least a job description 152a-n. For instance, if subject-specific data has a fuzzy set matching a job description fuzzy set by having a degree of overlap exceeding a threshold, processor 104 may classify the subject as being relevant or otherwise associated with the job description. Where multiple fuzzy matches are performed, degrees of
10 match for each respective fuzzy set may be computed and aggregated through, for instance, addition, averaging, or the like, to determine an overall degree of match.

Still referring to FIG. 5, in an embodiment, subject-specific data may be compared to multiple class fuzzy sets representing job-descriptions. For instance, subject-specific data may be represented by an individual fuzzy set that is compared to each of the multiple class fuzzy sets;
15 and a degree of overlap exceeding a threshold between the individual fuzzy set and any of the multiple class fuzzy sets may cause processor 104 to classify the subject as belonging to a job description. For instance, in one embodiment there may be two class fuzzy sets, representing a first job description and a second job description. First job description may have a first fuzzy set; second job description may have a second fuzzy set; and subject-specific data may have an
20 individual fuzzy set. Processor 104, for example, may compare an individual fuzzy set with each of first fuzzy set and second fuzzy set, as described above, and classify a subject to either, both, or neither of first job description nor second job description. Machine-learning methods as described throughout may, in a non-limiting example, generate coefficients used in fuzzy set equations as described above, such as without limitation x , c , and σ of a Gaussian set as
25 described above, as outputs of machine-learning methods. Likewise, subject-specific data may be used indirectly to determine a fuzzy set, as the fuzzy set may be derived from outputs of one or more machine-learning models that take the subject-specific data directly or indirectly as inputs. Although an exemplary application for fuzzy set matching is described above, fuzzy set matching may be used for any classifications or associations described within this disclosure.

Now referring to FIG. 6, an exemplary embodiment of method 600 for automated video record generation. At step 605, method 600 includes capturing, by plurality of sensors, at least audiovisual data from a user. Audiovisual data from a user may be based on an employment datum. An employment datum 120 may comprise of a user's work history. These steps may be
5 implemented as described above, without limitation, in FIGS. 1-5.

At step 610, method 600 generating, by plurality of sensors, at least an audiovisual datum. In one or more embodiments, audiovisual data produced by a user 108 may be as a result of a prompt 116 presented by processor 104. These steps may be implemented as described above, without limitation, in FIGS. 1-5.

10 At step 615, method 600 includes converting, by processor, the at least an audiovisual datum into a video file. In one or more embodiments, video file may be compressed in one or more methods discussed above.

At step 620, method 600 includes generating, by processor, a video record of a user based on the at least an audiovisual datum. Video record may be generated with a machine learning
15 algorithm discussed above. In an embodiment, video record 128 may include written and visual components. These steps may be implemented as described above, without limitation, in FIGS. 1-5.

At step 625, method 600 includes identifying, by processor, a plurality of temporal sections of the video file. At step 630, method 600 includes classifying, by processor, each
20 temporal section of the plurality of temporal sections to a resume prompt of a plurality of resume prompts, where the plurality of resume prompts is arranged in a prompt ordering. At step 635, method 600 includes assembling, by processor, the plurality of classified temporal sections into the video record using the prompt ordering.

In an embodiment, methods and systems described herein may perform or implement one
25 or more aspects of a cryptographic system. In one embodiment, a cryptographic system is a system that converts data from a first form, known as "plaintext," which is intelligible when viewed in its intended format, into a second form, known as "ciphertext," which is not intelligible when viewed in the same way. Ciphertext may be unintelligible in any format unless first converted back to plaintext. In one embodiment, a process of converting plaintext into
30 ciphertext is known as "encryption." Encryption process may involve the use of a datum, known

as an “encryption key,” to alter plaintext. Cryptographic system may also convert ciphertext back into plaintext, which is a process known as “decryption.” Decryption process may involve the use of a datum, known as a “decryption key,” to return the ciphertext to its original plaintext form. In embodiments of cryptographic systems that are “symmetric,” decryption key is
5 essentially the same as encryption key: possession of either key makes it possible to deduce the other key quickly without further secret knowledge. Encryption and decryption keys in symmetric cryptographic systems may be kept secret and shared only with persons or entities that the user of the cryptographic system wishes to be able to decrypt the ciphertext. One example of a symmetric cryptographic system is the Advanced Encryption Standard (“AES”),
10 which arranges plaintext into matrices and then modifies the matrices through repeated permutations and arithmetic operations with an encryption key.

In embodiments of cryptographic systems that are “asymmetric,” either encryption or decryption key cannot be readily deduced without additional secret knowledge, even given the possession of a corresponding decryption or encryption key, respectively; a common example is
15 a “public key cryptographic system,” in which possession of the encryption key does not make it practically feasible to deduce the decryption key, so that the encryption key may safely be made available to the public. An example of a public key cryptographic system is RSA, in which an encryption key involves the use of numbers that are products of very large prime numbers, but a decryption key involves the use of those very large prime numbers, such that deducing the
20 decryption key from the encryption key requires the practically infeasible task of computing the prime factors of a number which is the product of two very large prime numbers. Another example is elliptic curve cryptography, which relies on the fact that given two points P and Q on an elliptic curve over a finite field, and a definition for addition where $A + B = -R$, the point where a line connecting point A and point B intersects the elliptic curve, where “0,” the identity,
25 is a point at infinity in a projective plane containing the elliptic curve, finding a number k such that adding P to itself k times results in Q is computationally impractical, given correctly selected elliptic curve, finite field, and P and Q.

In some embodiments, systems and methods described herein produce cryptographic hashes, also referred to by the equivalent shorthand term “hashes.” A cryptographic hash, as used
30 herein, is a mathematical representation of a lot of data, such as files or blocks in a block chain

as described in further detail below; the mathematical representation is produced by a lossy “one-way” algorithm known as a “hashing algorithm.” Hashing algorithm may be a repeatable process; that is, identical lots of data may produce identical hashes each time they are subjected to a particular hashing algorithm. Because hashing algorithm is a one-way function, it may be impossible to reconstruct a lot of data from a hash produced from the lot of data using the hashing algorithm. In the case of some hashing algorithms, reconstructing the full lot of data from the corresponding hash using a partial set of data from the full lot of data may be possible only by repeatedly guessing at the remaining data and repeating the hashing algorithm; it is thus computationally difficult if not infeasible for a single computer to produce the lot of data, as the statistical likelihood of correctly guessing the missing data may be extremely low. However, the statistical likelihood of a computer or a set of computers simultaneously attempting to guess the missing data within a useful timeframe may be higher, permitting mining protocols as described in further detail below.

In an embodiment, hashing algorithm may demonstrate an “avalanche effect,” whereby even extremely small changes to lot of data produce drastically different hashes. This may thwart attempts to avoid the computational work necessary to recreate a hash by simply inserting a fraudulent datum in data lot, enabling the use of hashing algorithms for “tamper-proofing” data such as data contained in an immutable ledger as described in further detail below. This avalanche or “cascade” effect may be evinced by various hashing processes; persons skilled in the art, upon reading the entirety of this disclosure, will be aware of various suitable hashing algorithms for purposes described herein. Verification of a hash corresponding to a lot of data may be performed by running the lot of data through a hashing algorithm used to produce the hash. Such verification may be computationally expensive, albeit feasible, potentially adding up to significant processing delays where repeated hashing, or hashing of large quantities of data, is required, for instance as described in further detail below. Examples of hashing programs include, without limitation, SHA256, a NIST standard; further current and past hashing algorithms include Winternitz hashing algorithms, various generations of Secure Hash Algorithm (including “SHA-1,” “SHA-2,” and “SHA-3”), “Message Digest” family hashes such as “MD4,” “MD5,” “MD6,” and “RIPEMD,” Keccak, “BLAKE” hashes and progeny (e.g., “BLAKE2,” “BLAKE-256,” “BLAKE-512,” and the like), Message Authentication Code (“MAC”)-family

hash functions such as PMAC, OMAC, VMAC, HMAC, and UMAC, Poly1305-AES, Elliptic Curve Only Hash (“ECOH”) and similar hash functions, Fast-Syndrome-based (FSB) hash functions, GOST hash functions, the Grøstl hash function, the HAS-160 hash function, the JH hash function, the RadioGatún hash function, the Skein hash function, the Streebog hash function, the SWIFFT hash function, the Tiger hash function, the Whirlpool hash function, or any hash function that satisfies, at the time of implementation, the requirements that a cryptographic hash be deterministic, infeasible to reverse-hash, infeasible to find collisions, and have the property that small changes to an original message to be hashed will change the resulting hash so extensively that the original hash and the new hash appear uncorrelated to each other. A degree of security of a hash function in practice may depend both on the hash function itself and on characteristics of the message and/or digest used in the hash function. For example, where a message is random, for a hash function that fulfills collision-resistance requirements, a brute-force or “birthday attack” may to detect collision may be on the order of $O(2^{n/2})$ for n output bits; thus, it may take on the order of 2^{256} operations to locate a collision in a 512 bit output “Dictionary” attacks on hashes likely to have been generated from a non-random original text can have a lower computational complexity, because the space of entries they are guessing is far smaller than the space containing all random permutations of bits. However, the space of possible messages may be augmented by increasing the length or potential length of a possible message, or by implementing a protocol whereby one or more randomly selected strings or sets of data are added to the message, rendering a dictionary attack significantly less effective.

A “secure proof,” as used in this disclosure, is a protocol whereby an output is generated that demonstrates possession of a secret, such as device-specific secret, without demonstrating the entirety of the device-specific secret; in other words, a secure proof by itself, is insufficient to reconstruct the entire device-specific secret, enabling the production of at least another secure proof using at least a device-specific secret. A secure proof may be referred to as a “proof of possession” or “proof of knowledge” of a secret. Where at least a device-specific secret is a plurality of secrets, such as a plurality of challenge-response pairs, a secure proof may include an output that reveals the entirety of one of the plurality of secrets, but not all of the plurality of secrets; for instance, secure proof may be a response contained in one challenge-response pair. In an embodiment, proof may not be secure; in other words, proof may include a one-time

revelation of at least a device-specific secret, for instance as used in a single challenge-response exchange.

Secure proof may include a zero-knowledge proof, which may provide an output demonstrating possession of a secret while revealing none of the secret to a recipient of the output; zero-knowledge proof may be information-theoretically secure, meaning that an entity with infinite computing power would be unable to determine secret from output. Alternatively, zero-knowledge proof may be computationally secure, meaning that determination of secret from output is computationally infeasible, for instance to the same extent that determination of a private key from a public key in a public key cryptographic system is computationally infeasible.

Zero-knowledge proof algorithms may generally include a set of two algorithms, a prover algorithm, or “P,” which is used to prove computational integrity and/or possession of a secret, and a verifier algorithm, or “V” whereby a party may check the validity of P. Zero-knowledge proof may include an interactive zero-knowledge proof, wherein a party verifying the proof must directly interact with the proving party; for instance, the verifying and proving parties may be required to be online, or connected to the same network as each other, at the same time.

Interactive zero-knowledge proof may include a “proof of knowledge” proof, such as a Schnorr algorithm for proof on knowledge of a discrete logarithm. In a Schnorr algorithm, a prover commits to a randomness r , generates a message based on r , and generates a message adding r to a challenge c multiplied by a discrete logarithm that the prover is able to calculate; verification is performed by the verifier who produced c by exponentiation, thus checking the validity of the discrete logarithm. Interactive zero-knowledge proofs may alternatively or additionally include sigma protocols. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various alternative interactive zero-knowledge proofs that may be implemented consistently with this disclosure.

Alternatively, zero-knowledge proof may include a non-interactive zero-knowledge, proof, or a proof wherein neither party to the proof interacts with the other party to the proof; for instance, each of a party receiving the proof and a party providing the proof may receive a reference datum which the party providing the proof may modify or otherwise use to perform the proof. As a non-limiting example, zero-knowledge proof may include a succinct non-interactive arguments of knowledge (ZK-SNARKS) proof, wherein a “trusted setup” process creates proof

and verification keys using secret (and subsequently discarded) information encoded using a public key cryptographic system, a prover runs a proving algorithm using the proving key and secret information available to the prover, and a verifier checks the proof using the verification key; public key cryptographic system may include RSA, elliptic curve cryptography, ElGamal, or any other suitable public key cryptographic system. Generation of trusted setup may be performed using a secure multiparty computation so that no one party has control of the totality of the secret information used in the trusted setup; as a result, if any one party generating the trusted setup is trustworthy, the secret information may be unrecoverable by malicious parties.

As another non-limiting example, non-interactive zero-knowledge proof may include a Succinct Transparent Arguments of Knowledge (ZK-STARKS) zero-knowledge proof. In an embodiment, a ZK-STARKS proof includes a Merkle root of a Merkle tree representing evaluation of a secret computation at some number of points, which may be 1 billion points, plus Merkle branches representing evaluations at a set of randomly selected points of the number of points; verification may include determining that Merkle branches provided match the Merkle root, and that point verifications at those branches represent valid values, where validity is shown by demonstrating that all values belong to the same polynomial created by transforming the secret computation. In an embodiment, ZK-STARKS does not require a trusted setup.

Zero-knowledge proof may include any other suitable zero-knowledge proof. Zero-knowledge proof may include, without limitation bulletproofs. Zero-knowledge proof may include a homomorphic public-key cryptography (hPKC)-based proof. Zero-knowledge proof may include a discrete logarithmic problem (DLP) proof. Zero-knowledge proof may include a secure multi-party computation (MPC) proof. Zero-knowledge proof may include, without limitation, an incrementally verifiable computation (IVC). Zero-knowledge proof may include an interactive oracle proof (IOP). Zero-knowledge proof may include a proof based on the probabilistically checkable proof (PCP) theorem, including a linear PCP (LPCP) proof. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various forms of zero-knowledge proofs that may be used, singly or in combination, consistently with this disclosure.

In an embodiment, secure proof is implemented using a challenge-response protocol. In an embodiment, this may function as a one-time pad implementation; for instance, a

manufacturer or other trusted party may record a series of outputs (“responses”) produced by a device possessing secret information, given a series of corresponding inputs (“challenges”), and store them securely. In an embodiment, a challenge-response protocol may be combined with key generation. A single key may be used in one or more digital signatures as described in
5 further detail below, such as signatures used to receive and/or transfer possession of cryptocurrency assets; the key may be discarded for future use after a set period of time. In an embodiment, varied inputs include variations in local physical parameters, such as fluctuations in local electromagnetic fields, radiation, temperature, and the like, such that an almost limitless variety of private keys may be so generated. Secure proof may include encryption of a challenge
10 to produce the response, indicating possession of a secret key. Encryption may be performed using a private key of a public key cryptographic system, or using a private key of a symmetric cryptographic system; for instance, trusted party may verify response by decrypting an encryption of challenge or of another datum using either a symmetric or public-key cryptographic system, verifying that a stored key matches the key used for encryption as a
15 function of at least a device-specific secret. Keys may be generated by random variation in selection of prime numbers, for instance for the purposes of a cryptographic system such as RSA that relies prime factoring difficulty. Keys may be generated by randomized selection of parameters for a seed in a cryptographic system, such as elliptic curve cryptography, which is generated from a seed. Keys may be used to generate exponents for a cryptographic system such
20 as Diffie-Helman or ElGamal that are based on the discrete logarithm problem.

A “digital signature,” as used herein, includes a secure proof of possession of a secret by a signing device, as performed on provided element of data, known as a “message.” A message may include an encrypted mathematical representation of a file or other set of data using the private key of a public key cryptographic system. Secure proof may include any form of secure
25 proof as described above, including without limitation encryption using a private key of a public key cryptographic system as described above. Signature may be verified using a verification datum suitable for verification of a secure proof; for instance, where secure proof is enacted by encrypting message using a private key of a public key cryptographic system, verification may include decrypting the encrypted message using the corresponding public key and comparing the
30 decrypted representation to a purported match that was not encrypted; if the signature protocol is

well-designed and implemented correctly, this means the ability to create the digital signature is equivalent to possession of the private decryption key and/or device-specific secret. Likewise, if a message making up a mathematical representation of file is well-designed and implemented correctly, any alteration of the file may result in a mismatch with the digital signature; the
5 mathematical representation may be produced using an alteration-sensitive, reliably reproducible algorithm, such as a hashing algorithm as described above. A mathematical representation to which the signature may be compared may be included with signature, for verification purposes; in other embodiments, the algorithm used to produce the mathematical representation may be publicly available, permitting the easy reproduction of the mathematical representation
10 corresponding to any file.

In some embodiments, digital signatures may be combined with or incorporated in digital certificates. In one embodiment, a digital certificate is a file that conveys information and links the conveyed information to a “certificate authority” that is the issuer of a public key in a public key cryptographic system. Certificate authority in some embodiments contains data conveying
15 the certificate authority’s authorization for the recipient to perform a task. The authorization may be the authorization to access a given datum. The authorization may be the authorization to access a given process. In some embodiments, the certificate may identify the certificate authority. The digital certificate may include a digital signature.

In some embodiments, a third party such as a certificate authority (CA) is available to
20 verify that the possessor of the private key is a particular entity; thus, if the certificate authority may be trusted, and the private key has not been stolen, the ability of an entity to produce a digital signature confirms the identity of the entity and links the file to the entity in a verifiable way. Digital signature may be incorporated in a digital certificate, which is a document authenticating the entity possessing the private key by authority of the issuing certificate
25 authority and signed with a digital signature created with that private key and a mathematical representation of the remainder of the certificate. In other embodiments, digital signature is verified by comparing the digital signature to one known to have been created by the entity that purportedly signed the digital signature; for instance, if the public key that decrypts the known signature also decrypts the digital signature, the digital signature may be considered verified.

Digital signature may also be used to verify that the file has not been altered since the formation of the digital signature.

Now referring to FIG. 7, an apparatus for creating a video record is illustrated. Apparatus 700 includes a processor 704. Processor 704 may include any computing device as described in this disclosure, including without limitation a microcontroller, microprocessor, digital signal processor (DSP) and/or system on a chip (SoC) as described in this disclosure. Computing device may include, be included in, and/or communicate with a mobile device such as a mobile telephone or smartphone. Processor 704 may include a single computing device operating independently, or may include two or more computing device operating in concert, in parallel, sequentially or the like; two or more computing devices may be included together in a single computing device or in two or more computing devices. Processor 704 may interface or communicate with one or more additional devices as described below in further detail via a network interface device. Network interface device may be utilized for connecting processor 704 to one or more of a variety of networks, and one or more devices. Examples of a network interface device include, but are not limited to, a network interface card (*e.g.*, a mobile network interface card, a LAN card), a modem, and any combination thereof. Examples of a network include, but are not limited to, a wide area network (*e.g.*, the Internet, an enterprise network), a local area network (*e.g.*, a network associated with an office, a building, a campus or other relatively small geographic space), a telephone network, a data network associated with a telephone/voice provider (*e.g.*, a mobile communications provider data and/or voice network), a direct connection between two computing devices, and any combinations thereof. A network may employ a wired and/or a wireless mode of communication. In general, any network topology may be used. Information (*e.g.*, data, software etc.) may be communicated to and/or from a computer and/or a computing device. Processor 704 may include but is not limited to, for example, a computing device or cluster of computing devices in a first location and a second computing device or cluster of computing devices in a second location. Processor 704 may include one or more computing devices dedicated to data storage, security, distribution of traffic for load balancing, and the like. Processor 704 may distribute one or more computing tasks as described below across a plurality of computing devices of computing device, which may operate in parallel, in series, redundantly, or in any other manner used for distribution of tasks or

memory between computing devices. Processor 704 may be implemented using a “shared nothing” architecture in which data is cached at the worker, in an embodiment, this may enable scalability of apparatus 700 and/or computing device.

5 With continued reference to FIG. 7, processor 704 may be designed and/or configured to perform any method, method step, or sequence of method steps in any embodiment described in this disclosure, in any order and with any degree of repetition. For instance, processor 704 may be configured to perform a single step or sequence repeatedly until a desired or commanded outcome is achieved; repetition of a step or a sequence of steps may be performed iteratively and/or recursively using outputs of previous repetitions as inputs to subsequent repetitions, 10 aggregating inputs and/or outputs of repetitions to produce an aggregate result, reduction or decrement of one or more variables such as global variables, and/or division of a larger processing task into a set of iteratively addressed smaller processing tasks. Processor 704 may perform any step or sequence of steps as described in this disclosure in parallel, such as simultaneously and/or substantially simultaneously performing a step two or more times using 15 two or more parallel threads, processor cores, or the like; division of tasks between parallel threads and/or processes may be performed according to any protocol suitable for division of tasks between iterations. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which steps, sequences of steps, processing tasks, and/or data may be subdivided, shared, or otherwise dealt with using iteration, recursion, and/or parallel 20 processing. Apparatus 700 may include a memory 708. Memory 708 may be communicatively connected to processor 704 and may be configured to store information and/or datum related to apparatus 700. In one or more embodiments, memory 708 may be communicatively connected to a processor and configured to contain instructions configuring processor to execute any operations discussed in this disclosure. In one or more embodiments, memory 708 may include a 25 storage device, as described further in this disclosure below.

Processor 704 is configured to prompt a user with an initial request 712. As used in this disclosure, a “user” is an individual seeking employment. As used in this disclosure, a “request” is a request, such as a question, to a user for information about the user. A request may seek information about a user’s credentials, as described below. Processor 704 may be 30 communicatively connected to a user device 716 and transmit initial request 712 to the user

device 716. Processor 704 may prompt user with requests, as discussed below. Processor 704 may cause user device 716 to prompt user by, as non-limiting examples, audibly reciting the requests over speakers connected to user device 716 and/or visually displaying the requests on a screen of user device. As used in this disclosure, “communicatively connected” means connected
5 by way of a connection, attachment, or linkage between two or more relata which allows for reception and/or transmittance of information therebetween. For example, and without limitation, this connection may be wired or wireless, direct or indirect, and between two or more components, circuits, devices, systems, and the like, which allows for reception and/or transmittance of data and/or signal(s) therebetween. Data and/or signals therebetween may
10 include, without limitation, electrical, electromagnetic, magnetic, video, audio, radio and microwave data and/or signals, combinations thereof, and the like, among others. A communicative connection may be achieved, for example and without limitation, through wired or wireless electronic, digital or analog, communication, either directly or by way of one or more intervening devices or components. Further, communicative connection may include electrically
15 coupling or connecting at least an output of one device, component, or circuit to at least an input of another device, component, or circuit. For example, and without limitation, via a bus or other facility for intercommunication between elements of a computing device. Communicative connecting may also include indirect connections via, for example and without limitation, wireless connection, radio communication, low power wide area network, optical
20 communication, magnetic, capacitive, or optical coupling, and the like. In some instances, the terminology “communicatively coupled” may be used in place of communicatively connected in this disclosure.

As used in this disclosure, a “user device” is a computing device controlled and/or operated by a user. Computing device may be any computing device described in this disclosure,
25 such as a processor communicatively connected to a memory. User device 716 may be a personal computer such as a desktop, laptop, smart phone, and/or the like. Processor 704 may be configured to require information from user device 716, such as a username and a password, to verify the identity of user. Processor 704 may send verification to user, such as an email to user’s email address and/or a text message to user’s phone saved on a memory and/or database to which
30 processor 704 has access. Verification may include a link to click that sends a verification to

processor 704. Verification may include a temporary code for user to then input from user device 716 to confirm that the device is user device 716.

Processor 704 may be configured to receive at least a user document 720 from user device 716, memory, and/or a database such as a user document database. User document database may be implemented, without limitation, as a relational database, a key-value retrieval database such as a NOSQL database, or any other format or structure for use as a database that a person skilled in the art would recognize as suitable upon review of the entirety of this disclosure. User document database may alternatively or additionally be implemented using a distributed data storage protocol and/or data structure, such as a distributed hash table or the like.

User document database may include a plurality of data entries and/or records as described above. Data entries in a database may be flagged with or linked to one or more additional elements of information, which may be reflected in data entry cells and/or in linked tables such as tables related by one or more indices in a relational database. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which data entries in a database may store, retrieve, organize, and/or reflect data and/or records as used herein, as well as categories and/or populations of data consistently with this disclosure. As used in this disclosure, a “user document” is a document that includes information about the user and may include credentials 752 of user. As used in this disclosure, “credentials” are information about a user pertaining to qualifications of the user and may include, for example, skills, accreditations, a minimum grade point average (GPA), degree, major and/or focus of study, prior employment, experience, and/or the like. User document 720 may include a transcript from an educational institution the user attended such as a college transcript, a written resume, a video resume, a letter of recommendation, and/or the like. Initial request 712 may be based on at least a user document 720, such as user’s transcript and/or written resume. For example, processor 704 may be configured to utilize optical character recognition (OCR) or any word recognition process discussed in this disclosure to translate written text into machine-encoded text. Processor 704 may identify one or more keywords in user document 720. As used in this disclosure, a “keyword” is an element of word or syntax used to identify and/or match elements to each other. Keyword may include locations such as cities, states, and regions; credentials 752; criteria 732 as discussed below; and/or the like. Keywords may be stored in a database, such as keyword

database, from which processor 704 may retrieve the keywords. Keyword database may be implemented, without limitation, as a relational database, a key-value retrieval datastore such as a NOSQL database, or any other format or structure for use as a datastore that a person skilled in the art would recognize as suitable upon review of the entirety of this disclosure.

5 With continued reference to FIG. 7, processor 704 may be configured to recognize at least a keyword in user document 720 as a function of visual verbal content. In some cases, recognizing a plurality of keywords in user document 720 may include a language processing module. In some embodiments, samples, and examples of keywords submitted by a hiring entity or apparatus 700 administrator may be used to train language processing module in identifying
10 keywords. For instance, a sample may be a list of synonyms used for common words used by hiring entities, such as “advocate”, “attorney-at-law”, “counsel”, “counselor”, “counselor-at-law”, “lawyer”, and “legal eagle”. These samples and examples may additionally be used to classify keywords to similar keywords contained in a plurality of user identifiers, as described further below. Language processing module may include any hardware and/or software module.
15 For example, language processing module may be used to extract from user document 720 all information pertaining to “truck driver”.

 With continued reference to FIG. 7, identifying keywords in user document 720 may include matching a plurality of keywords to words in user document 720. In some embodiments, matching may include classifying keywords contained in keyword database to similar words
20 contained in user document 720. For example, keywords relating “web developer internship experience” may be matched to similar words in user document 720. Similar words may be based on synonyms of keywords as described above. Matching may occur through a classifier. A “classifier,” as used in this disclosure is a machine-learning model, such as a mathematical model, neural net, or program generated by a machine learning algorithm known as a
25 “classification algorithm,” as described in further detail below, that sorts inputs into categories or bins of data, outputting the categories or bins of data and/or labels associated therewith. Classifier may be configured to output at least a datum that labels or otherwise identifies a set of data that are clustered together, found to be close under a distance metric as described below, or the like. Processor 704 and/or another device may generate classifier using a classification
30 algorithm, defined as a processes whereby processor 704 derives classifier from training data. In

an embodiment, training data may include data from a database as described in this disclosure including keyword database, sample and examples of keywords and words similar to keywords, language processing module, and any other training data described throughout this disclosure. Classifier may take the plurality of keywords from keyword database as algorithm inputs.

- 5 Classifier may then use the training data disclosed above to output data bins of words similar to keywords matched to keywords. Each data bin may be categorized to each keyword and labeled with the keyword. Classification may be performed using, without limitation, linear classifiers such as without limitation logistic regression and/or naive Bayes classifiers, nearest neighbor classifiers such as k-nearest neighbors classifiers, support vector machines, least squares support
- 10 vector machines, fisher's linear discriminant, quadratic classifiers, decision trees, boosted trees, random forest classifiers, learning vector quantization, and/or neural network-based classifiers. Words similar to keywords that are classified and labeled according to the corresponding keywords may become keywords and/or be added to keyword database.

- Still referring to FIG. 7, processor 704 may be configured to generate classifier using a
- 15 Naïve Bayes classification algorithm. Naïve Bayes classification algorithm generates classifiers by assigning class labels to problem instances, represented as vectors of element values. Class labels are drawn from a finite set. Naïve Bayes classification algorithm may include generating a family of algorithms that assume that the value of a particular element is independent of the value of any other element, given a class variable. Naïve Bayes classification algorithm may be
- 20 based on Bayes Theorem expressed as $P(A/B) = P(B/A) P(A) \div P(B)$, where $P(A/B)$ is the probability of hypothesis A given data B also known as posterior probability; $P(B/A)$ is the probability of data B given that the hypothesis A was true; $P(A)$ is the probability of hypothesis A being true regardless of data also known as prior probability of A; and $P(B)$ is the probability of the data regardless of the hypothesis. A naïve Bayes algorithm may be generated by first
- 25 transforming training data into a frequency table. Processor 704 may then calculate a likelihood table by calculating probabilities of different data entries and classification labels. Processor 704 may utilize a naïve Bayes equation to calculate a posterior probability for each class. A class containing the highest posterior probability is the outcome of prediction. Naïve Bayes classification algorithm may include a gaussian model that follows a normal distribution. Naïve
- 30 Bayes classification algorithm may include a multinomial model that is used for discrete counts.

Naïve Bayes classification algorithm may include a Bernoulli model that may be utilized when vectors are binary.

With continued reference to FIG. 7, processor 704 may be configured to generate classifier using a K-nearest neighbors (KNN) algorithm. A “K-nearest neighbors algorithm” as used in this disclosure, includes a classification method that utilizes feature similarity to analyze how closely out-of-sample- features resemble training data to classify input data to one or more clusters and/or categories of features as represented in training data; this may be performed by representing both training data and input data in vector forms, and using one or more measures of vector similarity to identify classifications within training data, and to determine a classification of input data. K-nearest neighbors algorithm may include specifying a K-value, or a number directing the classifier to select the k most similar entries training data to a given sample, determining the most common classifier of the entries in a database, and classifying the known sample; this may be performed recursively and/or iteratively to generate classifier that may be used to classify input data as further samples. For instance, an initial set of samples may be performed to cover an initial heuristic and/or “first guess” at an output and/or relationship, which may be seeded, without limitation, using expert input received according to any process as described herein. As a non-limiting example, an initial heuristic may include a ranking of associations between inputs and elements of training data. Heuristic may include selecting some number of highest-ranking associations and/or training data elements.

With continued reference to FIG. 7, generating k-nearest neighbors algorithm may generate a first vector output containing a data entry cluster, generating a second vector output containing an input data, and calculate the distance between the first vector output and the second vector output using any suitable norm such as cosine similarity, Euclidean distance measurement, or the like. Each vector output may be represented, without limitation, as an n-tuple of values, where n is at least two values. Each value of n-tuple of values may represent a measurement or other quantitative value associated with a given category of data, or attribute, examples of which are provided in further detail below; a vector may be represented, without limitation, in n-dimensional space using an axis per category of value represented in n-tuple of values, such that a vector has a geometric direction characterizing the relative quantities of attributes in the n-tuple as compared to each other. Two vectors may be considered equivalent

where their directions, and/or the relative quantities of values within each vector as compared to each other, are the same; thus, as a non-limiting example, a vector represented as [5, 10, 15] may be treated as equivalent, for purposes of this disclosure, as a vector represented as [1, 2, 3].

Vectors may be more similar where their directions are more similar, and more different where their directions are more divergent; however, vector similarity may alternatively or additionally be determined using averages of similarities between like attributes, or any other measure of similarity suitable for any n-tuple of values, or aggregation of numerical similarity measures for the purposes of loss functions as described in further detail below. Any vectors as described herein may be scaled, such that each vector represents each attribute along an equivalent scale of values. Each vector may be “normalized,” or divided by a “length” attribute, such as a length attribute l as derived using a Pythagorean norm: $l = \sqrt{\sum_{i=0}^n a_i^2}$, where a_i is attribute number i of the vector. Scaling and/or normalization may function to make vector comparison independent of absolute quantities of attributes, while preserving any dependency on similarity of attributes; this may, for instance, be advantageous where cases represented in training data are represented by different quantities of samples, which may result in proportionally equivalent vectors with divergent values. Keyword may be consistent with disclosure of keyword as discussed further in this disclosure below..

Still referring to FIG. 7, in some embodiments, initial request 712 may be based on at least a keyword processor 704 identifies in user document 720. For example, processor 704 may identify “electrical engineering” in user’s university transcript and create an initial request 712 concerning electrical engineering such as, “Are you looking for a job in electrical engineering?” or “How many years of experience do you have working as an electrical engineer?” In some embodiments, user may select at least a keyword to indicate a type of job the user is interested in. For example, user may select “teacher” and a subcategory “kindergarten teacher” to identify that the user is seeking employment as a kindergarten teacher. Initial request 712 may include asking user for name, address, email address, phone number, whether they are currently employed, etc.

Still referring to FIG. 7, processor 704 may implement one or more algorithms or generate one or more machine-learning modules, such as initial request module 724, to generate initial request 712 to user. In one or more embodiments, the machine-learning module may be generated using training data. Training data may include inputs and corresponding predetermined

outputs so that a machine-learning module may use the correlations between the provided exemplary inputs and outputs to develop an algorithm and/or relationship that then allows the machine-learning module to determine its own outputs for inputs. Training data may contain correlations that a machine-learning process may use to model relationships between two or more categories of data elements. The exemplary inputs and outputs may come from a database, such as any database described in this disclosure, or be provided by a user such as a prospective employee, and/or an employer. In other embodiments, a machine-learning module may obtain a training set by querying a communicatively connected database that includes past inputs and outputs. Training data may include inputs from various types of databases, resources, and/or user inputs and outputs correlated to each of those inputs so that a machine-learning module may determine an output. Correlations may indicate causative and/or predictive links between data, which may be modeled as relationships, such as mathematical relationships, by machine-learning processes, as described in further detail below. In one or more embodiments, training data may be formatted and/or organized by categories of data elements by, for example, associating data elements with one or more descriptors corresponding to categories of data elements. As a non-limiting example, training data may include data entered in standardized forms by persons or processes, such that entry of a given data element in a given field in a form may be mapped to one or more descriptors of categories. Elements in training data may be linked to descriptors of categories by tags, tokens, or other data elements. Initial request module 724 may be generated using training data, such as initial request data. Initial request module 724 may be trained by correlated inputs and outputs of initial request data. Inputs of initial request data may include keywords and/or user documents 720 containing at least a keyword. Outputs of initial request data may include initial requests 712 corresponding to the inputs. Initial request data may be keywords and/or user documents 720 with keywords and corresponding initial requests 712 that have already been determined whether manually, by machine, or any other method. Initial request data may include previous outputs such that initial request module 724 iteratively produces outputs. Initial request module 724 using a machine-learning process may output initial request 712 based on input of at least a keyword and/or at least a user documents 720 containing at least a keyword and initial request data.

With continued reference to FIG. 7, processor 704 may be configured to receive a selection from user device 716 of a posting 728. A “posting,” as used in this disclosure, is a communication of a job position for which a prospective employer is seeking or may be seeking one or more candidates to potentially fill the job position. Processor 704 may receive a plurality of postings 728. A posting may include information about the employer such as the employer’s name and address; compensation such as a salary, an hourly wage, and/or benefits; a title of the job position; geographical location of where the job will be performed and/or whether the job is to be performed remotely; a description of the job position such as a listing of responsibilities, expectations, and/or goals to be accomplished; criteria 732; and/or the like. A job position may be part-time and/or full-time. Job position may be as an employee and/or contractor. As used in this disclosure, “criteria,” are skills, accreditations, a minimum grade point average (GPA), degree, major and/or focus of study, and/or experience. Criteria 732 may include requirements and/or preferences. As used in this disclosure, a “requirement” is a criterion that must be satisfied for a prospective employee to be eligible for consideration for a posting. As used in this disclosure, a “preference” is a desired criterion, but it is not required for a prospective employee to be considered for a posting. Initial request 712 may be based on a specific posting 728, such as a posting 728 for which user has selected to apply. For example, user may select posting 728 for a position as a professor in philosophy at a community college, and initial request 712 may be, without limitation, “How many years have you taught philosophy?”, “Have you written any published articles?”, or “Please explain your educational background in philosophy.” Processor 704 may be configured to analyze whether user document 720 addresses all criteria 732 of posting 728. Initial request 712 may be based on any criteria 732 not addressed in user document 720 or any criteria 732 not satisfactorily explained in the user document 720. For example, if a requirement of criteria 732 is five years’ experience in a managerial position and processor 704 does not have user document 720, user document 720 does not address an amount of managerial experience, or it is unclear from user document 720 whether user has at least five years of managerial experience, then initial request 712 may ask how many years of managerial experience the user has. Comparing user document 720 and posting 728 may be consistent with disclosure of comparing user datum and posting datum in as discussed further below in this disclosure,. In embodiments in which initial request 712 is based on posting 728, inputs of initial

request data may include criteria 732 and/or postings 728 containing criteria 732, and outputs of initial request data may include initial requests 712 corresponding to the inputs. Therefore, initial request module 724 may be configured to output initial request 712 based on posting 728 and/or criteria 732 and initial request data. Additional disclosure pertaining to posting is discussed
5 further below in this disclosure..

With continued reference to FIG. 7, processor 704 is be configured to determine the initial request. This may be done using initial request module 724. The initial request is determined based on posting 728. In some embodiments, processor 704 may be configured to determine the initial request based on a user document 720. In some embodiments, processor 704
10 may be configured to determine the initial request based on a keyword selected by the user.

Processor 704 may receive posting 728 from a computing device, store and/or retrieve from memory 708, and/or store and/or retrieve from a database such as a posting database. Posting database may be implemented, without limitation, as a relational database, a key-value retrieval database such as a NOSQL database, or any other format or structure for use as a
15 database that a person skilled in the art would recognize as suitable upon review of the entirety of this disclosure. Posting database may alternatively or additionally be implemented using a distributed data storage protocol and/or data structure, such as a distributed hash table or the like. Posting database may include a plurality of data entries and/or records as described above. Data entries in a database may be flagged with or linked to one or more additional elements of
20 information, which may be reflected in data entry cells and/or in linked tables such as tables related by one or more indices in a relational database. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which data entries in a database may store, retrieve, organize, and/or reflect data and/or records as used herein, as well as categories and/or populations of data consistently with this disclosure.

With continued reference to FIG. 7, processor 704 is configured to receive an initial response 736 from user. As used in this disclosure, an “initial response” is a response by user to an initial request 712. Processor 704 may receive initial response 736 from a computing device, such as user device 716, through a network, from a database, and/or store the initial response 736 in memory 708 and retrieve from the memory 708. Initial response 736 may be included in a
30 video record 740. Initial response 736 may include credentials 752. Initial response 736 may be

video record 740 of user responding to initial request 712, which may also include a video record 740 of the initial request 712. As used in this disclosure, a “video record” is data including an audio recording of a prospective employee for purposes of potentially acquiring a job. The audio recording may include verbal content 744. As used in this disclosure, “verbal content” is

5 comprehensible language-based communication. For example, verbal content 744 may include a monologue. Video record 740 may also include a visual recording of the prospective employee. Visual recording may include an image component 748. As used in this disclosure, “image component” may be a visual representation of information, such as a plurality of temporally sequential frames and/or pictures, related to video record 740. For example, image component

10 748 may include animations, still imagery, recorded video, and the like. In some cases, credentials 752 may be explicitly conveyed within video record 740. Alternatively, or additionally, in some cases, credentials 752 may be conveyed implicitly in video record 740. Video record 740 may be communicated by way of digital signals, for example between computing devices which are communicatively connected with at least a wireless network. Video

15 record 740 may be compressed to optimize speed and/or cost of transmission of video. Video record 740 may be compressed according to a video compression coding format (i.e., codec). Exemplary video compression codecs include H.26x codecs, MPEG formats, VVC, SVT-AV1, and the like. In some cases, compression of a digital video may be lossy, in which some information may be lost during compression. Alternatively, or additionally, in some cases,

20 compression of a video record 740 may be substantially lossless, where substantially no information is lost during compression. Processor 704 may receive posting 728 and/or video record 740 from a user, such as an employer, hiring agency, recruiting firm, and/or a prospective employee. Processor 704 may receive posting 728 and/or video record 740 from a computing device through a network, from a database, and or store posting 728 and/or video record 740 in a

25 memory and retrieve from the memory. Apparatus 700 may include a memory 708. Memory 708 may be communicatively connected to processor 704 and may be configured to store information and/or datum related to apparatus 700, such as initial request 712, subsequent request 756, posting 728 including criteria 732, user document 720, keywords selected by user, video record 740 including initial response 736 and subsequent response 764 with credentials 752.

Still referring to FIG. 7, processor 704 may be configured to extract a plurality of textual elements from video record 740 of initial response 736, which may include credentials 752. Processor 704 may include audiovisual speech recognition (AVSR) processes to recognize verbal content 744 in video records 740. For example, processor 704 may use image component 748 to aid in recognition of audible verbal content 744 such as viewing prospective employee move their lips to speak on video to process the audio content of video record 740. AVSR may use image component 748 to aid the overall translation of the audio verbal content 744 of video records 740. In some embodiments, AVSR may include techniques employing image processing capabilities in lip reading to aid speech recognition processes. In some cases, AVSR may be used to decode (i.e., recognize) indeterministic phonemes or help in forming a preponderance among probabilistic candidates. In some cases, AVSR may include an audio-based automatic speech recognition process and an image-based automatic speech recognition process. AVSR may combine results from both processes with feature fusion. Audio-based speech recognition process may analysis audio according to any method described herein, for instance using a Mel frequency cepstral coefficients (MFCCs) and/or log-Mel spectrogram derived from raw audio samples. Image-based speech recognition may perform feature recognition to yield an image vector. In some cases, feature recognition may include any feature recognition process described in this disclosure, for example a variant of a convolutional neural network. In some cases, AVSR employs both an audio datum and an image datum to recognize verbal content 744. For instance, audio vector and image vector may each be concatenated and used to predict speech made by prospective employee, who is 'on camera.'

With continued reference to FIG. 7, processor 704 is configured to analyze initial response 736. Analyzing initial response 736 may include identifying a plurality of credentials 752 from video record 740. In some cases, processor 704 may be configured to recognize at least a keyword as a function of visual verbal content 744. In some cases, recognizing at least keyword may include optical character recognition. In some cases, processor 704 may generate a transcript of much or even all verbal content 744 from video record 740. Processor 704 may use transcript to analyze the content of video record 740 and extract credentials 752.

Still refereeing to FIG. 7, in some embodiments, optical character recognition or optical character reader (OCR) includes automatic conversion of images of written (e.g., typed,

handwritten or printed text) into machine-encoded text. In some cases, recognition of at least a keyword from an image component 748 may include one or more processes, including without limitation optical character recognition (OCR), optical word recognition, intelligent character recognition, intelligent word recognition, and the like. In some cases, OCR may recognize
5 written text, one glyph or character at a time. In some cases, optical word recognition may recognize written text, one word at a time, for example, for languages that use a space as a word divider. In some cases, intelligent character recognition (ICR) may recognize written text one glyph or character at a time, for instance by employing machine-learning processes. In some cases, intelligent word recognition (IWR) may recognize written text, one word at a time, for
10 instance by employing machine-learning processes.

Still referring to FIG. 7, in some cases OCR may be an "offline" process, which analyses a static document or image frame. In some cases, handwriting movement analysis can be used as input to handwrite recognition. For example, instead of merely using shapes of glyphs and words, this technique may capture motions, such as the order in which segments are drawn, the
15 direction, and the pattern of putting the pen down and lifting it. This additional information may make handwriting recognition more accurate. In some cases, this technology may be referred to as "online" character recognition, dynamic character recognition, real-time character recognition, and intelligent character recognition.

Still referring to FIG. 7, in some cases, OCR processes may employ pre-processing of
20 image component 748. Pre-processing process may include without limitation de-skew, de-speckle, binarization, line removal, layout analysis or "zoning," line and word detection, script recognition, character isolation or "segmentation," and normalization. In some cases, a de-skew process may include applying a transform (e.g., homography or affine transform) to image component 748 to align text. In some cases, a de-speckle process may include removing positive
25 and negative spots and/or smoothing edges. In some cases, a binarization process may include converting an image from color or greyscale to black-and-white (i.e., a binary image). Binarization may be performed as a simple way of separating text (or any other desired image component) from a background of image component 748. In some cases, binarization may be required for example if an employed OCR algorithm only works on binary images. In some
30 cases, a line removal process may include removal of non-glyph or non-character imagery (e.g.,

boxes and lines). In some cases, a layout analysis or “zoning” process may identify columns, paragraphs, captions, and the like as distinct blocks. In some cases, a line and word detection process may establish a baseline for word and character shapes and separate words, if necessary. In some cases, a script recognition process may, for example in multilingual documents, identify script allowing an appropriate OCR algorithm to be selected. In some cases, a character isolation or “segmentation” process may separate signal characters, for example character-based OCR algorithms. In some cases, a normalization process may normalize aspect ratio and/or scale of image component 748.

Still referring to FIG. 7, in some embodiments an OCR process may include an OCR algorithm. Exemplary OCR algorithms include matrix matching process and/or feature extraction processes. Matrix matching may involve comparing an image to a stored glyph on a pixel-by-pixel basis. In some case, matrix matching may also be known as “pattern matching,” “pattern recognition,” and/or “image correlation.” Matrix matching may rely on an input glyph being correctly isolated from the rest of the image component 748. Matrix matching may also rely on a stored glyph being in a similar font and at a same scale as input glyph. Matrix matching may work best with typewritten text.

Still referring to FIG. 7, in some embodiments, an OCR process may include a feature extraction process. In some cases, feature extraction may decompose a glyph into at least a feature. Exemplary non-limiting features may include corners, edges, lines, closed loops, line direction, line intersections, and the like. In some cases, feature extraction may reduce dimensionality of representation and may make the recognition process computationally more efficient. In some cases, extracted feature may be compared with an abstract vector-like representation of a character, which might reduce to one or more glyph prototypes. General techniques of feature detection in computer vision are applicable to this type of OCR. In some

embodiments, machine-learning processes like nearest neighbor classifiers (e.g., k-nearest neighbors algorithm) may be used to compare image features with stored glyph features and choose a nearest match. OCR may employ any machine-learning process described in this disclosure, for example machine-learning processes described with reference to FIG. 2.

Exemplary non-limiting OCR software includes Cuneiform and Tesseract. Cuneiform is a multi-language, open-source optical character recognition system originally developed by Cognitive

Technologies of Moscow, Russia. Tesseract is free OCR software originally developed by Hewlett-Packard of Palo Alto, California, United States.

Still referring to FIG. 7, in some cases, OCR may employ a two-pass approach to character recognition. A first pass may try to recognize a character. Each character that is satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to recognize characters more accurately as it further analyzes image components 724. Since the adaptive classifier may have learned something useful a little too late to recognize characters on the first pass, a second pass is run over the image components 748. Second pass may include adaptive recognition and use characters recognized with high confidence on the first pass to recognize better remaining characters on the second pass. In some cases, two-pass approach may be advantageous for unusual fonts or low-quality image components 748 where visual verbal content 744 may be distorted. Another exemplary OCR software tool include OCRopus. OCRopus development is led by German Research Centre for Artificial Intelligence in Kaiserslautern, Germany. In some cases, OCR software may employ neural networks.

Still referring to FIG. 7, in some cases, OCR may include post-processing. For example, OCR accuracy may be increased, in some cases, if output is constrained by a lexicon. A lexicon may include a list or set of words that are allowed to occur in a document. In some cases, a lexicon may include, for instance, all the words in the English language, or a more technical lexicon for a specific field. In some cases, an output stream may be a plain text stream or file of characters. In some cases, an OCR process may preserve an original layout of visual verbal content 744. In some cases, near-neighbor analysis can make use of co-occurrence frequencies to correct errors, by noting that certain words are often seen together. For example, “Washington, D.C.” is generally far more common in English than “Washington DOC.” In some cases, an OCR process may make use of *a priori* knowledge of grammar for a language being recognized. For example, grammar rules may be used to help determine if a word is likely to be a verb or a noun. Distance conceptualization may be employed for recognition and classification. For example, a Levenshtein distance algorithm may be used in OCR post-processing to further optimize results.

With continued reference to FIG. 7, processor 704 may be configured to prompt user with a subsequent request 756 based on initial response 736. Subsequent request 756, similar to initial

request 712, may be based on posting 728, at least a user document 720, and/or at least a keyword selected by user. Subsequent request 756 may be a follow up to initial request 712. For example, if initial request 712 is “Are you open to relocating for employment?” and initial response 736 is in the affirmative such as “yes” or “I think so,” then subsequent request 756 may be “Which cities, states, or regions would you consider moving to?”. As another example, if user indicates in initial response 736 that the user was once employed as a manager of a team of people, subsequent request 756 may ask the user to describe a scenario where the user gave critical feedback to a team member when the team member fell below expectations. Subsequent request 756 may also be based on a specific posting 728 such as, “Would you be willing to relocate to Chicago?” if the posting 728 was for a position located in Chicago. Subsequent request 756 may additionally be based on user document 720. For example, subsequent request 756 may not ask user, “Would you be willing to relocate to Chicago?” if user document 720 shows that user lives in Chicago. As another example subsequent request 756 may be, “Tell me about your experience working for Acme, Inc.” if user document 720 states that user’s previous work experience includes working for Acme, Inc. Subsequent request 756 may seek additional information related to initial response 736. For example, if initial response 736 is an affirmative response to initial request 712 of whether user has written any published articles, subsequent request 756 may be, for example, “How many published articles have you authored?”, “What are the subject matters of the articles?”, “In which journals or publications were the articles published”, or “what are the titles of the articles?”.

Still referring to FIG. 7, processor 704 may implement or utilize one or more algorithms or generate one or more machine-learning modules, such as subsequent request module 760, to generate subsequent request 756 to user. Subsequent request module 760 may be generated using training data, such as subsequent request data. Subsequent request module 760 may be trained by correlated inputs and outputs of subsequent request data. Inputs of subsequent request data may include initial response 736, at least a keyword in initial response 736, initial request 712, and/or user document 720, or posting 728. Outputs of subsequent request data may include subsequent requests 760 corresponding to the inputs. Subsequent request data may be initial response 736, at least a keyword in initial response 736, initial request 712, and/or user document 720 and corresponding subsequent requests 760 that have already been determined whether manually, by

machine, or any other method. Subsequent request data may include previous outputs such that subsequent request module 760 iteratively produces outputs. Subsequent request module 760 using a machine-learning process may output subsequent request 756 based on input of initial response 736, at least a keyword in initial response 736, initial request 712, and/or user document 720 and subsequent request data. In some embodiments, processor 704 may be configured to determine the subsequent request. In some embodiments, this may be done using subsequent request module 760. In some embodiments, subsequent request 756 may be determined based on posting 728.

Initial request 712 and subsequent request 756 may be included in a set of requests for a subset of users. For example, each industry, type of profession, and/or field of work may have an associated set of requests to procure a consistent set of information from user. As example, Processor 704 may determine that user is an accountant by user document 720, selected keyword, and/or initial response 736. Processor 704 may then prompt user with a set of requests, such as initial request 712 and/or subsequent request 756, associated with accountant. Processor 704 may ask “Are you a Certified Public Accountant?” If user’s response is affirmative, processor 704 may ask user how many years of experience the user has as a CPA. Additional requests in the set of requests associated with accountants may include, for example, “What is your area of expertise?” and “Do you have experience in Accounts Receivable?” In some embodiments, processor 704 may be configured to determine which set of requests apply to user based on initial response 736. Processor 704 may determine a category corresponding to user. As used in this disclosure, a “category” is a profession and/or a field of work. For example, categories may include accountant, mechanical engineer, elementary school teacher, etc., and each category may have an associated set of requests. In these embodiments, inputs of subsequent request data may be initial responses 736, and outputs of subsequent request data may be categories corresponding to the initial responses 736, wherein the categories identify which set of requests apply to user. For example, categories may include accountant, mechanical engineer, elementary school teacher, etc., and each category may have an associated set of requests. Sets of requests may be stored in memory 708 or a database.

Processor 704 may be configured to receive a subsequent response 764 from user. As used in this disclosure, an “subsequent response” is a response by user to a subsequent request.

Processor 704 may receive subsequent response 764 from a computing device, such as user device 716, through a network, from a database, and/or store the subsequent response 764 in memory 708 and retrieve from the memory 708. Subsequent response 764 may be included in video record 740. Subsequent response 764 may include credentials 752. Subsequent response 5 764 may be video record 740 of user responding to subsequent request 756, which may also include video record 740 of the subsequent request 756.

With continued reference to FIG. 7, processor 704 may be configured to determine a score 768 based on video record 740 including initial response 736 and subsequent response 764. Score 768 may reflect user's conversational ability. Score 768 may be based on verbal content 10 744, such as how often user said filler words. As used in this disclosure, a "filler word" is a word that is used to fill a lull in speech. Filler words may include "um," "like", "uh", and the like. For example, processor 704 may provide a 700 score 768 if user said no filler words in initial response 736 and subsequent response 764. Score 768 may be based on a duration of the initial response 736 and a duration of the subsequent response 764. For example, use of five filler 15 words in initial response 736 of thirty-five seconds may receive a higher score 768 and use of five filler words in initial response 736 of twelve seconds.

Processor 704 may create video record 740 as a function of initial response 736 and subsequent response 764. Initial response 736 and subsequent response 764 may include a visual component. Video record 740 may include an audio recording of initial response 736 and 20 subsequent response 764. In some embodiments, video record 740 may include all of initial response 736 and subsequent response 764. Video record 740 may include initial request 712 and subsequent request 756 so the context of initial response 736 and subsequent response 764 are preserved. Video record 740 may include just segments of initial response 736 and/or subsequent response 764 and exclude portions that are not substantive, such as pauses and camera 25 adjustments.

Referring now to FIG. 8, an exemplary embodiment of an immutable sequential listing 800 is illustrated. Data elements are listing in immutable sequential listing 800; data elements may include any form of data, including textual data, image data, encrypted data, cryptographically hashed data, and the like. Data elements may include, without limitation, one 30 or more at least a digitally signed assertion. In one embodiment, a digitally signed assertion 804

is a collection of textual data signed using a secure proof as described in further detail below; secure proof may include, without limitation, a digital signature. Collection of textual data may contain any textual data, including without limitation American Standard Code for Information Interchange (ASCII), Unicode, or similar computer-encoded textual data, any alphanumeric data, punctuation, diacritical mark, or any character or other marking used in any writing system to convey information, in any form, including any plaintext or cyphertext data; in an embodiment, collection of textual data may be encrypted, or may be a hash of other data, such as a root or node of a Merkle tree or hash tree, or a hash of any other information desired to be recorded in some fashion using a digitally signed assertion 804. In an embodiment, collection of textual data states that the owner of a certain transferable item represented in a digitally signed assertion 804 register is transferring that item to the owner of an address. A digitally signed assertion 804 may be signed by a digital signature created using the private key associated with the owner's public key, as described above.

Still referring to FIG. 8, in some embodiments, an address is a textual datum identifying the recipient of virtual currency or another item of value, such as initial response 736 and subsequent response 764, in a digitally signed assertion 804. In some embodiments, address may be linked to a public key, the corresponding private key of which is owned by the recipient of a digitally signed assertion 804. For instance, address may be the public key. Address may be a representation, such as a hash, of the public key. Address may be linked to the public key in memory of a computing device, for instance via a "wallet shortener" protocol. Where address is linked to a public key, a transferee in a digitally signed assertion 804 may record a subsequent a digitally signed assertion 804 transferring some or all of the value transferred in the first a digitally signed assertion 804 to a new address in the same manner. A digitally signed assertion 804 may contain textual information that is not a transfer of some item of value in addition to, or as an alternative to, such a transfer. For instance, as described in further detail below, a digitally signed assertion 804 may indicate a confidence level associated with a distributed storage node as described in further detail below.

In an embodiment, and still referring to FIG. 8 immutable sequential listing 800 records a series of at least a posted content in a way that preserves the order in which the at least a posted content took place. Temporally sequential listing may be accessible at any of various security

settings; for instance, and without limitation, temporally sequential listing may be readable and modifiable publicly, may be publicly readable but writable only by entities and/or devices having access privileges established by password protection, confidence level, or any device authentication procedure or facilities described herein, or may be readable and/or writable only
5 by entities and/or devices having such access privileges. Access privileges may exist in more than one level, including, without limitation, a first access level or community of permitted entities and/or devices having ability to read, and a second access level or community of permitted entities and/or devices having ability to write; first and second community may be overlapping or non-overlapping. In an embodiment, posted content and/or immutable sequential
10 listing 800 may be stored as one or more zero knowledge sets (ZKS), Private Information Retrieval (PIR) structure, or any other structure that allows checking of membership in a set by querying with specific properties. Such database may incorporate protective measures to ensure that malicious actors may not query the database repeatedly in an effort to narrow the members of a set to reveal uniquely identifying information of a given posted content.

15 Still referring to FIG. 8, immutable sequential listing 800 may preserve the order in which the at least a posted content took place by listing them in chronological order; alternatively or additionally, immutable sequential listing 800 may organize digitally signed assertions 804 into sub-listings 808 such as “blocks” in a blockchain, which may be themselves collected in a temporally sequential order; digitally signed assertions 804 within a sub-listing 808
20 may or may not be temporally sequential. User document 720, Posting 728, initial response 736, subsequent response, video record 740, and/or score 768 may be posted on immutable sequential listing 800, such as blockchain. Training data for any machine-learning module discussed in this disclosure may be posted on immutable sequential listing 800, such as blockchain. A master list may be included. Master list may include a hash-table and/or distributed hash table which may
25 be used to locate a requestor-linked data store. For example, a public key associated with a requestor containing location information pertaining to requestor-linked data store may be converted into a series of hash functions. This may occur by converting an entry into a series of integers by using a hash function. A hash function may include any function that may be used to map a set of data which falls into the hash table. Hash functions may be stored in a hash table,
30 where it can be quickly retrieved using a hashed key. The hashed key may then be used to access

requestor-linked data store when prompted. Using the hashed key, a hash function may compute an index that may suggest where requestor-linked data store may be found. Locating may also be performed by linking the at least an encrypted data record to a digital signature associated with the requestor. Requestor may produce a digital signature, which may then be linked to the at least
5 an encrypted data record and locate to the location of the at least an encrypted data record. When the digital signature is presented, it may contain location information of the at least an encrypted data record and allow access control regulator to locate the precise location of encrypted data record. For example, digital signature may be generated using a public and/or private key linked to requestor which may contain location information of encrypted data record. In an
10 embodiment, encrypted data record may be linked to a requestor key, so that when a requestor key is presented, location of encrypted data record becomes apparent. Locating may also be performed by information that may be contained in data access request. For example, a data access request associated with a user may contain location information of encrypted data record that requestor is attempting to access. When generating a data access request, requestor may
15 specify the location of encrypted data record that may then be transmitted to access control regulator. Additional disclosure pertaining to immutable sequential listing can be found further in this disclosure.

With continued reference to FIG. 8, the ledger may preserve the order in which at least a posted content took place by listing them in sub-listings 808 and placing the sub-listings 808 in
20 chronological order. The immutable sequential listing 800 may be a distributed, consensus-based ledger, such as those operated according to the protocols promulgated by Ripple Labs, Inc., of San Francisco, Calif., or the Stellar Development Foundation, of San Francisco, Calif., or of Thunder Consensus. In some embodiments, the ledger is a secured ledger; in one embodiment, a secured ledger is a ledger having safeguards against alteration by unauthorized parties. The
25 ledger may be maintained by a proprietor, such as a system administrator on a server, that controls access to the ledger; for instance, the user account controls may allow contributors to the ledger to add at least a posted content to the ledger, but may not allow any users to alter at least a posted content that have been added to the ledger. In some embodiments, ledger is
cryptographically secured; in one embodiment, a ledger is cryptographically secured where each
30 link in the chain contains encrypted or hashed information that makes it practically infeasible to

alter the ledger without betraying that alteration has taken place, for instance by requiring that an administrator or other party sign new additions to the chain with a digital signature. Immutable sequential listing 800 may be incorporated in, stored in, or incorporate, any suitable data structure, including without limitation any database, datastore, file structure, distributed hash table, directed acyclic graph or the like. In some embodiments, the timestamp of an entry is cryptographically secured and validated via trusted time, either directly on the chain or indirectly by utilizing a separate chain. In one embodiment the validity of timestamp is provided using a time stamping authority as described in the RFC 3161 standard for trusted timestamps, or in the ANSI ASC x9.95 standard. In another embodiment, the trusted time ordering is provided by a group of entities collectively acting as the time stamping authority with a requirement that a threshold number of the group of authorities sign the timestamp. Immutable sequential listing 800 and/or any component of the immutable sequential listing 800, such as sub-listing 808 and digitally signed assertions 804, may be validated by processor 704 as discussed further in this disclosure.

In some embodiments, and with continued reference to FIG. 8, immutable sequential listing 800, once formed, may be inalterable by any party, no matter what access rights that party possesses. For instance, immutable sequential listing 800 may include a hash chain, in which data is added during a successive hashing process to ensure non-repudiation. Immutable sequential listing 800 may include a block chain. In one embodiment, a block chain is immutable sequential listing 800 that records one or more new at least a posted content in a data item known as a sub-listing 808 or "block." An example of a block chain is the BITCOIN block chain used to record BITCOIN transactions and values. Sub-listings 808 may be created in a way that places the sub-listings 808 in chronological order and link each sub-listing 808 to a previous sub-listing 808 in the chronological order so that any computing device may traverse the sub-listings 808 in reverse chronological order to verify any at least a posted content listed in the block chain. Each new sub-listing 808 may be required to contain a cryptographic hash describing the previous sub-listing 808. In some embodiments, the block chain may contain a single first sub-listing 808 sometimes known as a "genesis block."

Still referring to FIG. 8, the creation of a new sub-listing 808 may be computationally expensive; for instance, the creation of a new sub-listing 808 may be designed by a "proof of

work” protocol accepted by all participants in forming the immutable sequential listing 800 to take a powerful set of computing devices a certain period of time to produce. Where one sub-listing 808 takes less time for a given set of computing devices to produce the sub-listing 808, protocol may adjust the algorithm to produce the next sub-listing 808 so that it will require more steps; where one sub-listing 808 takes more time for a given set of computing devices to produce the sub-listing 808, protocol may adjust the algorithm to produce the next sub-listing 808 so that it will require fewer steps. As an example, protocol may require a new sub-listing 808 to contain a cryptographic hash describing its contents; the cryptographic hash may be required to satisfy a mathematical condition, achieved by having the sub-listing 808 contain a number, called a nonce, whose value is determined after the fact by the discovery of the hash that satisfies the mathematical condition. Continuing the example, the protocol may be able to adjust the mathematical condition so that the discovery of the hash describing a sub-listing 808 and satisfying the mathematical condition requires more or less steps, depending on the outcome of the previous hashing attempt. Mathematical condition, as an example, might be that the hash contains a certain number of leading zeros and a hashing algorithm that requires more steps to find a hash containing a greater number of leading zeros, and fewer steps to find a hash containing a lesser number of leading zeros. In some embodiments, production of a new sub-listing 808 according to the protocol is known as “mining.” The creation of a new sub-listing 808 may be designed by a “proof of stake” protocol as will be apparent to those skilled in the art upon reviewing the entirety of this disclosure.

Continuing to refer to FIG. 8, in some embodiments, protocol also creates an incentive to mine new sub-listings 808. The incentive may be financial; for instance, successfully mining a new sub-listing 808 may result in the person or entity that mines the sub-listing 808 receiving a predetermined amount of currency. The currency may be fiat currency. Currency may be cryptocurrency as defined below. In other embodiments, incentive may be redeemed for particular products or services; the incentive may be a gift certificate with a particular business, for instance. In some embodiments, incentive is sufficiently attractive to cause participants to compete for the incentive by trying to race each other to the creation of sub-listings 808. Each sub-listing 808 created in immutable sequential listing 800 may contain a record or at least a

posted content describing one or more addresses that receive an incentive, such as virtual currency, as the result of successfully mining the sub-listing 808.

With continued reference to FIG. 8, where two entities simultaneously create new sub-listings 808, immutable sequential listing 800 may develop a fork; protocol may determine which
5 of the two alternate branches in the fork is the valid new portion of the immutable sequential listing 800 by evaluating, after a certain amount of time has passed, which branch is longer. “Length” may be measured according to the number of sub-listings 808 in the branch. Length may be measured according to the total computational cost of producing the branch. Protocol may treat only at least a posted content contained in the valid branch as valid at least a posted
10 content. When a branch is found invalid according to this protocol, at least a posted content registered in that branch may be recreated in a new sub-listing 808 in the valid branch; the protocol may reject “double spending” at least a posted content that transfer the same virtual currency that another at least a posted content in the valid branch has already transferred. As a result, in some embodiments the creation of fraudulent at least a posted content requires the
15 creation of a longer immutable sequential listing 800 branch by the entity attempting the fraudulent at least a posted content than the branch being produced by the rest of the participants; as long as the entity creating the fraudulent at least a posted content is likely the only one with the incentive to create the branch containing the fraudulent at least a posted content, the computational cost of the creation of that branch may be practically infeasible, guaranteeing the
20 validity of all at least a posted content in the immutable sequential listing 800.

Still referring to FIG. 8, additional data linked to at least a posted content may be incorporated in sub-listings 808 in the immutable sequential listing 800; for instance, data may be incorporated in one or more fields recognized by block chain protocols that permit a person or computer forming a at least a posted content to insert additional data in the immutable sequential
25 listing 800. In some embodiments, additional data is incorporated in an unspendable at least a posted content field. For instance, the data may be incorporated in an OP_RETURN within the BITCOIN block chain. In other embodiments, additional data is incorporated in one signature of a multi-signature at least a posted content. In an embodiment, a multi-signature at least a posted content is at least a posted content to two or more addresses. In some embodiments, the two or
30 more addresses are hashed together to form a single address, which is signed in the digital

signature of the at least a posted content. In other embodiments, the two or more addresses are concatenated. In some embodiments, two or more addresses may be combined by a more complicated process, such as the creation of a Merkle tree or the like. In some embodiments, one or more addresses incorporated in the multi-signature at least a posted content are typical crypto-
5 currency addresses, such as addresses linked to public keys as described above, while one or more additional addresses in the multi-signature at least a posted content contain additional data related to the at least a posted content; for instance, the additional data may indicate the purpose of the at least a posted content, aside from an exchange of virtual currency, such as the item for which the virtual currency was exchanged. In some embodiments, additional information may
10 include network statistics for a given node of network, such as a distributed storage node, e.g. the latencies to nearest neighbors in a network graph, the identities or identifying information of neighboring nodes in the network graph, the trust level and/or mechanisms of trust (e.g. certificates of physical encryption keys, certificates of software encryption keys, (in non-limiting example certificates of software encryption may indicate the firmware version, manufacturer,
15 hardware version and the like), certificates from a trusted third party, certificates from a decentralized anonymous authentication procedure, and other information quantifying the trusted status of the distributed storage node) of neighboring nodes in the network graph, IP addresses, GPS coordinates, and other information informing location of the node and/or neighboring nodes, geographically and/or within the network graph. In some embodiments, additional
20 information may include history and/or statistics of neighboring nodes with which the node has interacted. In some embodiments, this additional information may be encoded directly, via a hash, hash tree or other encoding.

Now referring to FIG. 9, an exemplary embodiment of a method 900 for creating a video record is illustrated. At step 905, processor prompts a user with initial request; this may be
25 implemented, without limitation, as described above in reference to FIGS. 1-8. Prompting the user with initial request includes determining the initial request based on a posting. Processor may determine initial request based on user document. Processor may determine initial request based on keyword selected by user.

At step 910, processor receives initial response from user; this may be implemented,
30 without limitation, as described above in reference to FIGS. 1-8.

At step 915, processor prompts user with subsequent request based on initial response; this may be implemented, without limitation, as described above in reference to FIGS. 1-8.

Processor may generate subsequent request by utilizing a machine-learning module. In some embodiments, processor may determine and/or generate the subsequent request based on a posting. Processor may determine category corresponding to user. Subsequent request may be associated with category.

At step 920, processor receives subsequent response from user; this may be implemented, without limitation, as described above in reference to FIGS. 1-8. Initial response and subsequent response may include visual component.

At step 925, processor creates video record as a function of initial response and subsequent response; this may be implemented, without limitation, as described above in reference to FIGS. 1-8. Video record may include audio recording of initial response and subsequent response. In some embodiments, the processor may be configured to determine a score based on a duration of the initial response and a duration of the subsequent response.

Now referring to FIG. 10, an apparatus for matching video records with job posting using audiovisual data processing is illustrated. Apparatus 1000 includes a processor 1004. Processor 1004 may include any computing device as described in this disclosure, including without limitation a microcontroller, microprocessor, digital signal processor (DSP) and/or system on a chip (SoC) as described in this disclosure. Computing device may include, be included in, and/or communicate with a mobile device such as a mobile telephone or smartphone. Processor 1004 may include a single computing device operating independently, or may include two or more computing device operating in concert, in parallel, sequentially or the like; two or more computing devices may be included together in a single computing device or in two or more computing devices. Processor 1004 may interface or communicate with one or more additional devices as described below in further detail via a network interface device. Network interface device may be utilized for connecting processor 1004 to one or more of a variety of networks, and one or more devices. Examples of a network interface device include, but are not limited to, a network interface card (*e.g.*, a mobile network interface card, a LAN card), a modem, and any combination thereof. Examples of a network include, but are not limited to, a wide area network (*e.g.*, the Internet, an enterprise network), a local area network (*e.g.*, a network associated with

an office, a building, a campus or other relatively small geographic space), a telephone network, a data network associated with a telephone/voice provider (e.g., a mobile communications provider data and/or voice network), a direct connection between two computing devices, and any combinations thereof. A network may employ a wired and/or a wireless mode of

5 communication. In general, any network topology may be used. Information (e.g., data, software etc.) may be communicated to and/or from a computer and/or a computing device. Processor 1004 may include but is not limited to, for example, a computing device or cluster of computing devices in a first location and a second computing device or cluster of computing devices in a second location. Processor 1004 may include one or more computing devices dedicated to data
10 storage, security, distribution of traffic for load balancing, and the like. Processor 1004 may distribute one or more computing tasks as described below across a plurality of computing devices of computing device, which may operate in parallel, in series, redundantly, or in any other manner used for distribution of tasks or memory between computing devices. Processor 1004 may be implemented using a “shared nothing” architecture in which data is cached at the
15 worker, in an embodiment, this may enable scalability of apparatus 1000 and/or computing device.

With continued reference to FIG. 10, processor 1004 may be designed and/or configured to perform any method, method step, or sequence of method steps in any embodiment described in this disclosure, in any order and with any degree of repetition. For instance, processor 1004
20 may be configured to perform a single step or sequence repeatedly until a desired or commanded outcome is achieved; repetition of a step or a sequence of steps may be performed iteratively and/or recursively using outputs of previous repetitions as inputs to subsequent repetitions, aggregating inputs and/or outputs of repetitions to produce an aggregate result, reduction or decrement of one or more variables such as global variables, and/or division of a larger
25 processing task into a set of iteratively addressed smaller processing tasks. Processor 1004 may perform any step or sequence of steps as described in this disclosure in parallel, such as simultaneously and/or substantially simultaneously performing a step two or more times using two or more parallel threads, processor cores, or the like; division of tasks between parallel threads and/or processes may be performed according to any protocol suitable for division of
30 tasks between iterations. Persons skilled in the art, upon reviewing the entirety of this disclosure,

will be aware of various ways in which steps, sequences of steps, processing tasks, and/or data may be subdivided, shared, or otherwise dealt with using iteration, recursion, and/or parallel processing.

Still referring to FIG. 10, processor 1004 is configured to receive a posting 1008
5 including criteria 1012. Processor 1004 may be communicatively connected to a network, as discussed above. “Communicatively connected,” for the purposes of this disclosure, is a process whereby one device, component, or circuit is able to receive data from and/or transmit data to another device, component, or circuit. Communicative connection may be performed by wired or wireless electronic communication, either directly or by way of one or more intervening devices
10 or components. In an embodiment, communicative connection includes electrically connection an output of one device, component, or circuit to an input of another device, component, or circuit. Communicative connection may be performed via a bus or other facility for intercommunication between elements of a computing device. Communicative connection may include indirect connections via “wireless” connection, low power wide area network, radio
15 communication, optical communication, magnetic, capacitive, or optical connection, or the like. In an embodiment, communicative connecting may include electrically connecting an output of one device, component, or circuit to an input of another device, component, or circuit. Communicative connecting may be performed via a bus or other facility for intercommunication between elements of a computing device. Communicative connecting may include indirect
20 connections via “wireless” connection, low power wide area network, radio communication, optical communication, magnetic, capacitive, or optical connection, or the like.

With continued reference to FIG. 10, network may include one or more other processors. A “posting,” as used in this disclosure, is a communication of a job position for which a prospective employer is seeking or may be seeking one or more candidates to potentially fill the
25 job position. Processor 1004 may receive a plurality of postings 1008. A posting may include information about the employer such as the employer’s name and address; compensation such as a salary, an hourly wage, and/or benefits; a title of the job position; geographical location of where the job will be performed and/or whether the job is to be performed remotely; a description of the job position such as a listing of responsibilities, expectations, and/or goals to
30 be accomplished; criteria; and/or the like. A job position may be part-time and/or full-time. Job

position may be as an employee and/or contractor. As used in this disclosure, “criteria,” are skills, accreditations, a minimum grade point average (GPA), degree, major and/or focus of study, and/or experience. Criteria 1012 may include requirements and/or preferences. As used in this disclosure, a “requirement” is a criterion that must be satisfied for a prospective employee to
5 be eligible for consideration for a posting. As used in this disclosure, a “preference” is a desired criterion, but it is not required for a prospective employee to be considered for a posting.

With continued reference to FIG. 10, processor 1004 is configured to receive a video record 1016 including credentials 1028. As used in this disclosure, a “video record” is data including an audio recording of a prospective employee for purposes of potentially acquiring a
10 job. The audio recording may include verbal content 1020. As used in this disclosure, “verbal content” is comprehensible language-based communication. For example, verbal content 1020 may include a monologue. Video record 1016 may also include a visual recording of the prospective employee. Visual recording may include an image component 1024. As used in this disclosure, “image component” may be a visual representation of information, such as a plurality
15 of temporally sequential frames and/or pictures, related to video record 1016. For example, image component 1024 may include animations, still imagery, recorded video, and the like. As used in this disclosure, “credentials” are information about a prospective employee pertaining to qualifications of the prospective employee and may include, for example, skills, accreditations, a minimum grade point average (GPA), degree, major and/or focus of study, prior employment,
20 experience, and/or the like. In some cases, credentials 1028 may be explicitly conveyed within video record 1016. Alternatively, or additionally, in some cases, credentials 1028 may be conveyed implicitly in video record 1016. Video record 1016 may be communicated by way of digital signals, for example between computing devices which are communicatively connected with at least a wireless network. Video record 1016 may be compressed to optimize speed and/or
25 cost of transmission of video. Video record 1016 may be compressed according to a video compression coding format (i.e., codec). Exemplary video compression codecs include H.26x codecs, MPEG formats, VVC, SVT-AV1, and the like. In some cases, compression of a digital video may be lossy, in which some information may be lost during compression. Alternatively, or additionally, in some cases, compression of a video record 1016 may be substantially lossless,
30 where substantially no information is lost during compression. Processor 1004 may receive

posting 1008 and/or video record 1016 from a user, such as an employer, hiring agency, recruiting firm, and/or a prospective employee. Processor 1004 may receive posting 1008 and/or video record 1016 from a computing device through a network, from a database, and or store posting 1008 and/or video record 1016 in a memory and retrieve from the memory. Apparatus 1000 may include a memory 1040. Memory 1040 may be communicatively connected to processor 1004 and may be configured to store information and/or datum related to apparatus 1000, such as posting 1008 including criteria 1012 and/or video record 1016 including credentials 1028. In one or more embodiments, memory 1040 may be communicatively connected to a processor and configured to contain instructions configuring processor to determine compatibility score 1032. In one or more embodiments, memory component may include a storage device, as described further in this disclosure below.

Still referring to FIG. 10, processor 1004 is configured to extract a plurality of textual elements 1040 from video record 1016, which may include credentials 1028. Processor 1004 may include audiovisual speech recognition (AVSR) processes to recognize verbal content 1020 in video records 1016. For example, processor 1004 may use image component 1024 to aid in recognition of audible verbal content 1020 such as viewing prospective employee move their lips to speak on video to process the audio content of video record 1016. AVSR may use image component 1024 to aid the overall translation of the audio verbal content 1020 of video records 1016. In some embodiments, AVSR may include techniques employing image processing capabilities in lip reading to aid speech recognition processes. In some cases, AVSR may be used to decode (i.e., recognize) indeterministic phonemes or help in forming a preponderance among probabilistic candidates. In some cases, AVSR may include an audio-based automatic speech recognition process and an image-based automatic speech recognition process. AVSR may combine results from both processes with feature fusion. Audio-based speech recognition process may analysis audio according to any method described herein, for instance using a Mel frequency cepstral coefficients (MFCCs) and/or log-Mel spectrogram derived from raw audio samples. Image-based speech recognition may perform feature recognition to yield an image vector. In some cases, feature recognition may include any feature recognition process described in this disclosure, for example a variant of a convolutional neural network. In some cases, AVSR employs both an audio datum and an image datum to recognize verbal content 1020. For

instance, audio vector and image vector may each be concatenated and used to predict speech made by prospective employee, who is ‘on camera.’

Still referring to FIG. 10, processor 1004 is configured to identify a plurality of credentials 1028 from video record 1016. In some cases, processor 1004 may be configured to recognize at least a keyword as a function of visual verbal content 1020. In some cases, recognizing at least keyword may include optical character recognition. As used in this disclosure, a “keyword” is an element of word or syntax used to identify and/or match elements to each other. At least a keyword may include credentials 1028. In some cases, processor 1004 may generate a transcript of much or even all verbal content 1020 from video record 1016. Processor 1004 may use transcript to analyze the content of video record 1016 and extract credentials 1028.

With continued reference to FIG. 10, processor 1004 is configured to generate a compatibility score 1032 for video record 1016 based on criteria 1012 and credentials 1028. Processor 1004 may compare transcript of video record 1016 to posting 1008 to generate compatibility score 1032. In one or more embodiments, processor 1004 may implement a compatibility algorithm or generate a compatibility machine-learning module, such as machine-learning module 1036, to determine a compatibility score 1032 as a measurement of compatibility between credentials 1028 in video record 1016 and criteria 1012 in posting 1008. For the purposes of this disclosure, a “compatibility score” is a measurable value representing a relevancy of credentials 1028 to criteria 1012. In one or more non-limiting embodiments, compatibility score 1032 may be a quantitative characteristic, such as a numerical value within a set range. For example, a compatibility score may be a “2” for a set range of 1-10, where “1” represents the compatibility between credentials 1028 and criteria 1012, and thus the compatibility between a prospective employee and a posting 1008, having a minimum compatibility and “10” represents credentials 1028 and criteria 1012 having a maximum compatibility. In other non-limiting embodiments, compatibility score 1032 may be a quality characteristic, such as a color coding, where each color is associated with a level of compatibility. In one or more embodiments, if a compatibility score 1032 is “low”, then a prospective employee and a job position are considered to have a minimum compatibility; if a compatibility score 1032 is “high”, then prospective employee and posting 1008 are considered

to have a maximum compatibility. Credentials 1028 may be validated as described in U.S. Patent Application No. 17/486,461 filed on September 27, 2021, and entitled “SYSTEMS AND METHODS FOR SCORE GENERATION FOR APPLICANT TRACKING”, the entirety of which is incorporated herein by reference. Compatibility score 1032 may be combined with
5 and/or aggregated with other scores as described, for instance, in U.S. Patent Application No. 17/486,461.

Still referring to FIG. 10, machine-learning module 1036 may use a classifier, which may include any classifier as described in this disclosure.

Processor 1004 may use a supervised machine-learning model to generate a compatibility
10 score 1032 given an input of a plurality of criteria 1012 and an input of corresponding a plurality of credentials 1028. Inputs may be manually inputted and/or labeled to identify which of the criteria 1012 correspond to which of the credentials 1028, causing the machine to learn correlations between the criteria 1012 and credentials 1028 that match for a high compatibility score 1032. Processor 1004 may compute compatibility score 1032 associated with each pairing
15 of criteria 1012 and credentials 1028 and select pairings to minimize and/or maximize the score, depending on whether an optimal result is represented, respectively, by a minimal and/or maximal score; a mathematical function, described herein as an “objective function,” may be used by processor 1004 to score each possible pairing. Processor 1004 may pair credentials 1028 of video records 1016 with criteria 1012 of postings 1008 that optimizes compatibility score
20 1032. Generation of objective function may include generation of a function to score and weight factors to achieve compatibility score 1032 for each feasible pairing. In some embodiments, pairings may be scored in a matrix for optimization, where columns represent criteria 1012 of various postings 1008 and rows represent credentials 1028 of various video records 1016 potentially paired therewith; each cell of such a matrix may represent a score of a pairing of the
25 corresponding criteria 1012 to the corresponding credentials 1028.

With continued reference to FIG. 10, matching a video to a posting to optimize an objective function may include performing a greedy algorithm process. A “greedy algorithm” is defined as an algorithm that selects locally optimal choices, which may or may not generate a globally optimal solution. For instance, computing device 1004 may select pairings so that
30 compatibility scores 1032 associated therewith are the best score for each video record 1016

and/or for each posting 1008. In such an example, optimization may determine the combination of postings 1008 such that each pairing includes the highest score possible.

Still referring to FIG. 10, objective function may be formulated as a linear objective function. Which processor 1004 may solve using a linear program such as without limitation a mixed-integer program. A “linear program,” as used in this disclosure, is a program that optimizes a linear objective function, given at least a constraint. For instance, and without limitation, objective function may seek to maximize a total score $\sum_{V \in R} \sum_{S \in S} c_{vp} x_{vp}$, where V is the set of all video records 1016 v , S is a set of all postings p , c_{vp} is a score of a pairing of a given posting with a given video, and x_{vp} is 1 if a video v is paired with posting p , and 0 otherwise. Continuing the example, constraints may specify that each posting 1008 is assigned to only one video record 1016, and each video record 1016 is assigned only one posting 1008. A mathematical solver may be implemented to solve for the set of feasible pairings that maximizes the sum of scores across all pairings; mathematical solver may implemented on processor 1004 and/or another device in apparatus 1000, and/or may be implemented on third-party solver.

With continued reference to FIG. 10, optimizing objective function may include minimizing a loss function, where a “loss function” is an expression an output of which an optimization algorithm minimizes to generate an optimal result. As a non-limiting example, processor 1004 may assign variables relating to a set of parameters, which may correspond to score components as described above, calculate an output of mathematical expression using the variables, and select a pairing that produces an output having the lowest size, according to a given definition of “size,” of the set of outputs representing each of plurality of candidate ingredient combinations; size may, for instance, included absolute value, numerical size, or the like. Selection of different loss functions may result in identification of different potential pairings as generating minimal outputs. Objectives represented in an objective function and/or loss function may include minimization of delivery times. Objectives may include the highest possible compatibility score 1032 for each posting 1008 on an individual basis, the highest possible compatibility score 1032 for each video record 1016 on an individual basis, and/or the highest average compatibility scores 1032 across all postings 1008.

Still referring to FIG. 10, in some embodiments, processor 1004 may query a keyword with a text search. Keyword may include words relating to skills such as C++, Java, Computer

Aided Design (CAD), welding, Excel, etc. Keyword may include education background such as Master of Science (MS), Bachelor of Science (BS), Juris Doctor (JD), and the like. Text search may include techniques for searching a single computer-stored document or a collection of documents, for example in a database. Text search may include full-text search. Full-text search may be distinguished from searches based on metadata or on field-based searching (e.g., fields such as titles, abstracts, selected sections, or bibliographical references). In an exemplary full-text search, processor 1004 may examine all words in every stored document as it tries to match search criteria (for example, keywords). Alternatively, a text search may be limited to fields, such as with field-based searching.

With continued reference to FIG. 10, in some embodiments, text searching may include querying. Database may be implemented, without limitation, as a relational database, a key-value retrieval database such as a NOSQL database, or any other format or structure for use as a database that a person skilled in the art would recognize as suitable upon review of the entirety of this disclosure. Database may alternatively or additionally be implemented using a distributed data storage protocol and/or data structure, such as a distributed hash table or the like. Database may include a plurality of data entries and/or records as described above. Data entries in a database may be flagged with or linked to one or more additional elements of information, which may be reflected in data entry cells and/or in linked tables such as tables related by one or more indices in a relational database. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which data entries in a database may store, retrieve, organize, and/or reflect data and/or records as used herein, as well as categories and/or populations of data consistently with this disclosure. In some cases, querying of at least a video element may include any number of querying tools, including without limitation keywords (as described above), field-restricted search, Boolean queries, phrase search, concept search, concordance search, proximity search, regular expression, fuzzy search, wildcard search, and the like. In some cases, keywords may be used to perform a query. In some cases, a document (or trained indexers) may supply a list of words that describe subject of the document, including without limitation synonyms of words that describe the subject. In some cases, keywords may improve recall, for instance if the keyword list includes a keyword that is not in text of a document. In some cases, querying tools may include field-restricted search. A field-restricted

search may allow a queries scope to be limited to within a particular field within a stored data record, such as "Title" or "Author." In some cases, a query tool may include Boolean queries. Searches that use Boolean operators (for example, "encyclopedia" AND "online" NOT "Encarta") can dramatically increase precision of a search. In some cases, an AND operator may say, in effect, "Do not retrieve any document unless it contains both of these terms." In some cases, a NOT operator may say, in effect, "Do not retrieve any document that contains this word." In some cases, a retrieval list retrieving too few documents, may prompt and OR operator to be used in place of an AND operator to increase recall; consider, for example, "encyclopedia" AND "online" OR "Internet" NOT "Encarta." This search will retrieve documents about online encyclopedias that use the term "Internet" instead of "online." In some cases, search precision and recall are interdependent and negatively correlated in text searching. In some cases, a query tool may include phrase search. In some cases, a phrase search may match only those documents that contain a specified phrase. In some cases, a query tool may include a concept search. In some cases, a concept search may be based on multi-word concepts, for example compound term processing. In some cases, a query tool may include a concordance search. In some cases, a concordance search may produce an alphabetical list of all principal words that occur in a text and may include their immediate context. In some cases, a query tool may include a proximity search. In some cases, a proximity search matches only those documents that contain two or more words that are separated by a specified number of words, are in the same sentence, or an in the same paragraph. A query tool may include a regular expression. In some cases, a regular expression may employ a complex but powerful querying syntax that can be used to specify retrieval conditions with precision, for instance database syntax. A query tool may include a fuzzy search. In some cases, a fuzzy search may search for a document that matches given terms while allowing for some variation around them. In some cases, a query tool may include a wildcard search. In some cases, a wildcard search may substitute one or more characters in a search query for a wildcard character such as an asterisk. For example, using a wildcard, such as an asterisk, in a search query "s*n" will search for terms inclusive of "sin," "son," "sun," and the like.

Continuing to reference FIG. 10, as a function of the comparison result, a ranking of video records 1016 may be generated. In some cases, generating the ranking of video records

1016 may include linear regression techniques. In some cases, comparison result may be presented as a percentage of relevance between video record 1016 and the posting 1008.

Processor 1004 may be designed and configured to create a machine-learning model using techniques for development of linear regression models. Linear regression models may include

5 ordinary least squares regression, which aims to minimize the square of the difference between predicted outcomes and actual outcomes according to an appropriate norm for measuring such a difference (e.g. a vector-space distance norm); coefficients of the resulting linear equation may be modified to improve minimization. Linear regression models may include ridge regression methods, where the function to be minimized includes the least-squares function plus term

10 multiplying the square of each coefficient by a scalar amount to penalize large coefficients.

Linear regression models may include least absolute shrinkage and selection operator (LASSO) models, in which ridge regression is combined with multiplying the least-squares term by a factor of 1 divided by double the number of samples. Linear regression models may include a multi-task lasso model wherein the norm applied in the least-squares term of the lasso model is

15 the Frobenius norm amounting to the square root of the sum of squares of all terms. Linear regression models may include the elastic net model, a multi-task elastic net model, a least angle regression model, a LARS lasso model, an orthogonal matching pursuit model, a Bayesian regression model, a logistic regression model, a stochastic gradient descent model, a perceptron model, a passive aggressive algorithm, a robustness regression model, a Huber regression model,

20 or any other suitable model that may occur to persons skilled in the art upon reviewing the entirety of this disclosure. Linear regression models may be generalized in an embodiment to polynomial regression models, whereby a polynomial equation (e.g. a quadratic, cubic or higher-order equation) providing a best predicted output/actual output fit is sought; similar methods to those described above may be applied to minimize error functions, as will be apparent to persons skilled in the art upon reviewing the entirety of this disclosure.

25 With continued reference to FIG. 10, for the purposes of this disclosure, a “compatibility algorithm” is an algorithm that determines the relevancy of a prospective employee’s characteristics, according to credentials 1028 in video record 1016, with qualifications of a job position, according to criteria 1012 in posting 1008. Compatibility algorithm may include machine-learning processes that are used to calculate one or more compatibility scores 1032.

Machine-learning process may be trained by using training data associated with past calculations and corresponding credentials 1028 and criteria 1012. Compatibility score 1032 may be determined by, for example, if a certain numerical value and/or percentage of criteria 1012 are satisfied by credentials 1028, where the more employment position data that matches user data, the higher the score and the greater the compatibility between prospective employee and posting 1008. For example, and without limitation, criteria 1012 of posting 1008 may include a qualification of requiring a teacher with at least five years of work experience, and credentials 1028 in video record 1016 of the teacher may include seven years of work experience in teaching, then a numerical value representing compatibility score 1032 may be increased due to the data correlating, thus indicating that prospective employee has a high compatibility for the job position. In some embodiments, processor 1004 may distinguish between criteria 1012 that are requirements and criteria 1012 that are preferences. For example, a greater weight, and therefore a greater impact on compatibility score 1032, may be given to credentials 1028 that match criteria 1012 that are requirements than credentials 1028 that match criteria 1012 that are preferences. In some embodiments, compatibility score 1032 may include a compatibility score for requirements, a compatibility score for preferences, and/or a comprehensive compatibility score for all criteria 1012. In some embodiments, compatibility score 1032 for video record 1016 with credentials 1028 that do not satisfy all criteria 1012 of posting 1008 that are requirements may be zero, a non-score, and/or otherwise distinguished from video records 1016 with credentials 1028 that do satisfy all criteria 1012 that are requirements. As used in this disclosure, a criterion being “satisfied” means that one or more credentials meets or exceeds the criterion. For example, a criterion requiring five years’ experience in a given field is satisfied by a credential of having worked six years in the given field. Whether credentials 1028 satisfy criteria 1012 may be determined by an algorithm discussed in this disclosure such as compatibility algorithm, a machine-learning process discussed in this disclosure, and/or the like applied to posting 1008. For example, dates, numbers, and/or words describing lengths of time may be Keywords that are identified, processor 1004 may calculate the length of time described if not facially apparent, processor 1004 may determine from neighboring text the significance of the period of time, which may include identifying neighboring Keywords. If processor 1004 determines that criteria 1012 includes an amount of time of experience in a field, then processor

1004 may use the same algorithm and/or machine-learning process to identify information in video record 1016 addressing the criteria 1012 and analyze whether credentials 1028 satisfy the criteria 1012. In an embodiment, compatibility algorithm may be received from a remote device. In some embodiments, compatibility algorithm is generated by processor 1004. In one or more
5 embodiments, compatibility algorithm may be generated as a function of credentials 1028 and/or criteria 1012.

 In one or more embodiments, a machine-learning process may be used to determine compatibility algorithm or to generate a machine-learning model that may directly calculate compatibility score 1032. In one or more embodiments, a machine-learning model may be
10 generated using training data. Training data may include inputs and corresponding predetermined outputs so that a machine-learning module may use the correlations between the provided exemplary inputs and outputs to develop an algorithm and/or relationship that then allows the machine-learning module to determine its own outputs for inputs. Training data may contain correlations that a machine-learning process may use to model relationships between two or
15 more categories of data elements. The exemplary inputs and outputs may come from a database, such as any database described in this disclosure, or be provided by a user, such as a prospective employee and/or an employer. In other embodiments, a machine-learning module may obtain a training set by querying a communicatively connected database that includes past inputs and outputs. Training data may include inputs from various types of databases, resources, and/or user
20 inputs and outputs correlated to each of those inputs so that a machine-learning module may determine an output, such as compatibility score 1032, for an input, such as criteria 1012 and credentials 1028. Training data may be obtained from and/or in the form of previous posting-video record matches. Previous video record matches may include resumes such as video resumes and written resumes. Correlations may indicate causative and/or predictive links
25 between data, which may be modeled as relationships, such as mathematical relationships, by machine-learning processes, as described in further detail below. In one or more embodiments, training data may be formatted and/or organized by categories of data elements by, for example, associating data elements with one or more descriptors corresponding to categories of data elements. As a non-limiting example, training data may include data entered in standardized
30 forms by persons or processes, such that entry of a given data element in a given field in a form

may be mapped to one or more descriptors of categories. Elements in training data may be linked to descriptors of categories by tags, tokens, or other data elements.

Still referring to FIG. 10, processor 1004 is configured to match video record 1016 with posting 1008 based on compatibility score 1032. The match may be automated. Processor 1004 may receive a plurality of video records 1016 and a plurality of postings 1008. Processor 1004 may match one or more video records 1016 to posting 1008 that have the highest compatibility scores 1032 with the posting 1008. Processor 1004 may match a predetermined amount of video records 1016 such as 1, 5, 10, 20, or any other amount to each posting 1008. Processor 1004 may only match video records 1016 that obtain at least a minimum determined compatibility score 1032. Processor 1004 may provide corresponding compatibility scores 1032 for each matched video record 1016. Processor 1004 may provide transcript for each corresponding video record 1016 that matches. Processor 1004 may rank the matched video records 1016. Processor 1004 may identify any criteria 1012 that was not satisfied by credentials 1028 of matched video record 1016. Processor 1004 may identify credentials 1028 that exceed criteria 1012, such as additional degrees obtained, a higher GPA than required, additional years of experience, and/or the like. In some embodiments, video record 1016 will not match with posting 1008 if credentials 1028 do not satisfy all criteria 1012 of posting 1008. Matches may be based on an amount of previous matches of video record 1016 with other postings. For example, if processor 1004 has already matched a first video record of video record 1016 with a predetermined amount of postings, such as 1, 5, 10, 20, or any other amount of postings, then processor 1004 may not match first video 1016 resume with posting 1008. Processor 1004 may prevent first posting 1008 from matching if previous matches were within a specified amount of time.

Now referring to FIG. 11, an exemplary embodiment of a method 1100 for matching video records with postings using audiovisual data processing is illustrated. At step 1105, a processor receives a posting including a plurality of criteria; this may be implemented, without limitation, as described above in reference to FIGS. 1-10. Criteria may have one or more requirements and one or more preferences.

At step 1110, processor receives a video record; this may be implemented, without limitation, as described above in reference to FIGS. 1-10. Video record may include an audio recording. Processor may be configured to generate a transcript of video record. Video record

may contain visual verbal content, and processor may be configured to use image processing to convert visual verbal content to textual data and compare textual data to the posting.

At step 1115, processor extracts a plurality of textual elements from video record; this may be implemented, without limitation, as described above in reference to FIGS. 1-10. At step 1120, processor identifies a plurality of credentials from video record. At step 1125, processor generates a compatibility score for video record based on plurality of criteria and plurality of credentials; this may be implemented, without limitation, as described above in reference to FIGS. 1-10. Requirement may be weighted greater than preference for the compatibility score. Processor may be configured to compare transcript to posting.

At step 1130, processor matches video record with posting based on compatibility score; this may be implemented, without limitation, as described above in reference to FIGS. 1-10. The match may be based on whether credentials satisfy requirement. Match may be based on an amount of previous matches of video record. Match may be based on previous matches of the video record within a specified of time.

Referring now to FIG. 12, an exemplary embodiment of apparatus 1200 for parsing and comparing resume video duplications is illustrated. Apparatus 1200 includes computing device 1204. Computing device 1204 may include any computing device as described in this disclosure, including without limitation a microcontroller, microprocessor, digital signal processor (DSP) and/or system on a chip (SoC) as described in this disclosure. Computing device 1204 may include, be included in, and/or communicate with a mobile device such as a mobile telephone or smartphone. Computing device 1204 may include a single computing device operating independently, or may include two or more computing device operating in concert, in parallel, sequentially or the like; two or more computing devices may be included together in a single computing device or in two or more computing devices. Computing device 1204 may interface or communicate with one or more additional devices as described below in further detail via a network interface device. Network interface device may be utilized for connecting computing device 1204 to one or more of a variety of networks, and one or more devices. Examples of a network interface device include, but are not limited to, a network interface card (e.g., a mobile network interface card, a LAN card), a modem, and any combination thereof. Examples of a network include, but are not limited to, a wide area network (e.g., the Internet, an enterprise

network), a local area network (e.g., a network associated with an office, a building, a campus or other relatively small geographic space), a telephone network, a data network associated with a telephone/voice provider (e.g., a mobile communications provider data and/or voice network), a direct connection between two computing devices, and any combinations thereof. A network
5 may employ a wired and/or a wireless mode of communication. In general, any network topology may be used. Information (e.g., data, software etc.) may be communicated to and/or from a computer and/or a computing device. Computing device 1204 may include but is not limited to, for example, a computing device or cluster of computing devices in a first location and a second computing device or cluster of computing devices in a second location. Computing
10 device 1204 may include one or more computing devices dedicated to data storage, security, distribution of traffic for load balancing, and the like. Computing device 1204 may distribute one or more computing tasks as described below across a plurality of computing devices of computing device, which may operate in parallel, in series, redundantly, or in any other manner used for distribution of tasks or memory between computing devices. Computing device 1204
15 may be implemented using a “shared nothing” architecture in which data is cached at the worker, in an embodiment, this may enable scalability of apparatus 1200 and/or computing device.

With continued reference to FIG. 12, computing device 1204 may be designed and/or configured to perform any method, method step, or sequence of method steps in any embodiment described in this disclosure, in any order and with any degree of repetition. For instance,
20 computing device 1204 may be configured to perform a single step or sequence repeatedly until a desired or commanded outcome is achieved; repetition of a step or a sequence of steps may be performed iteratively and/or recursively using outputs of previous repetitions as inputs to subsequent repetitions, aggregating inputs and/or outputs of repetitions to produce an aggregate result, reduction or decrement of one or more variables such as global variables, and/or division
25 of a larger processing task into a set of iteratively addressed smaller processing tasks. Computing device 1204 may perform any step or sequence of steps as described in this disclosure in parallel, such as simultaneously and/or substantially simultaneously performing a step two or more times using two or more parallel threads, processor cores, or the like; division of tasks between parallel threads and/or processes may be performed according to any protocol suitable for division of
30 tasks between iterations. Persons skilled in the art, upon reviewing the entirety of this disclosure,

will be aware of various ways in which steps, sequences of steps, processing tasks, and/or data may be subdivided, shared, or otherwise dealt with using iteration, recursion, and/or parallel processing.

Still referring to FIG. 12, computing device 1204 is configured to acquire a plurality of video elements from a target video resume 1208. As used in this disclosure, “video elements” are diverse types of features from a video resume such as image features, frame features, sound features, graphical features, and the like. As used in this disclosure, a “video resume” is a video in visual and/or audio form to provide a recording promoting a jobseeker. In some cases, video resume 1212 may include content that is representative or communicative of at least attribute 1224 of subject 1220. As used in this disclosure, a “subject” is a person, for example a jobseeker. Subject 1220 may be represented directly by video resume 1212. For example, in some cases, image component 1216a may include an image of subject 1220. As used in this disclosure, an “image component” may be a visual representation of information, such as a plurality of temporally sequential frames and/or pictures, related to video resume and target video resume. For example, image component 1216a-b may include animations, still imagery, recorded video, and the like. Attributes 1224 may include subject’s 1220 skills, competencies, credentials, talents, and the like. In some cases, attributes 1224 may be explicitly conveyed within video resume 1212. Alternatively, or additionally, in some cases, attributes 1224 may be conveyed implicitly with video resume 1212.

As used in this disclosure, a “target video resume” includes at least image component 1216b that may be selected based on a request by a user or job-applicant from one or more databases in which multiple video resumes 1212 of users or job-applicants are stored. Target video resume 1208 may be the basis for computing device 1204 to search for other video resumes 1212 based on target resume 1208 video properties. Video elements from target video 1208 may relate to diverse types of features of target video resume 1208. As used in this disclosure, a “feature” is an individually measurable property or characteristic such as image feature, frame feature, sound feature, graphical feature, and textual feature. These features may be in the form of a floating-point number feature, a binarized feature, a fence feature, a recall feature, a pooling feature, a reranking feature, and the like. Target video resume 1208 may be communicated by way of digital signals, for example between computing devices which are

communicatively connected with at least a wireless network. Digital video may be compressed to optimize speed and/or cost of transmission of video. Videos may be compressed according to a video compression coding format (i.e., codec). Exemplary video compression codecs include H.26x codecs, MPEG formats, VVC, SVT-AV1, and the like. In some cases, compression of a digital video may be lossy, in which some information may be lost during compression.

Alternatively, or additionally, in some cases, compression of a digital video may be substantially lossless, where substantially no information is lost during compression.

Still referring to FIG. 12, Video resume 1212 may be representative subject-specific data. As used in this disclosure, “subject-specific data” is any element of information that is associated with a specific subject. Exemplary forms of subject-specific data include image component 1216a-b, video resume 1212, non-verbal content 1232, verbal content 1236, audio component, as well as any information derived directly or indirectly from video resume 1212 or any other subject-specific data. For example, subject-specific data could be the physical properties of subject 1220, such as their body posture or facial expression. Subject-specific data could also be audio sensory properties of subject 1220, such as tone of voice or background audio in a resume video 1212.

In some cases, video resume 1212 may include non-verbal content 1232. As used in this disclosure, “non-verbal content” is all communication that is not characterized as verbal content. As used in this disclosure, “verbal content” is comprehensible language-based communication. For example, verbal content may include “visual verbal content” which is literal and/or written verbal content. Non-verbal content 1232 includes all forms of communication which are not conveyed with use of language. Exemplary non-verbal content may include change in intonation and/or stress in a speaker’s voice, expression of emotion, and the like. For example, in some cases, non-verbal content may include visual non-verbal content. As used in this disclosure, “visual non-verbal content” is non-verbal content 1232 that is visually represented. In some cases, visual non-verbal content may be included within video resume 1212 by way of image component 1216a.

In some cases, a non-verbal classifier may classify non-verbal content 1232 present in one or more image component 1216a to one or more of target video resume 1208, a feature.

Non-verbal classifier may include a number of classifiers, for example each being tasked with

classifying a particular attribute 1224 or form of non-verbal content 1232. For example, in some cases, non-verbal classifier may classify a video resume 1212 and related subject 1220 as associated with a feature representative of ‘personable.’ Non-verbal classifier may include another specialized visual non-verbal classifier to classify visual non-verbal content as appearing
5 ‘personable’ that is, for example, as having appropriate posture, facial expressions, manner of dress, and the like. In some cases, classifier may include or a constituent part of tree structure, for making associations based upon video resume.

With continued reference to FIG. 12, in some embodiments, image component 1216a-b may include or otherwise represent verbal content 1236. For instance, written or visual verbal
10 content may be included within image component 1216a-b. Visual verbal content may include images of written text represented by image component 1216a-b. For example, visual verbal content may include, without limitation, digitally generated graphics, images of written text (e.g., typewritten, and the like), signage, and the like.

Still referring to FIG. 12, in some embodiments, image component 1216a-b may include
15 or otherwise represent audible verbal content related to at least an attribute 1224 of subject 1220. As used in this disclosure, “audible verbal content” is oral (e.g., spoken) verbal content. In some cases, audible verbal content may be included within video resume 1212 by way of an audio component. As used in this disclosure, an “audio component” is a representation of audio, for example a sound, a speech, and the like. In some cases, verbal content 1236 may be related to at
20 least an attribute of subject. Additionally, or alternatively, visual verbal content and audible verbal content may be used as inputs to classifiers as described throughout this disclosure.

In some cases, computing device 1204 may include audiovisual speech recognition (AVSR) processes to recognize verbal content 1236 in video resumes 1212. For example, computing device 1204 may use image content 1216a-b to aid in recognition of audible verbal
25 content such as viewing subject 1220 move their lips to speak on video to process the audio content of video resume 1212. AVSR may use image component 1216a-b to aid the overall translation of the audio verbal content of video resumes 1212. In some embodiments, AVSR may include techniques employing image processing capabilities in lip reading to aid speech recognition processes. In some cases, AVSR may be used to decode (i.e., recognize)
30 indeterministic phonemes or help in forming a preponderance among probabilistic candidates. In

some cases, AVSR may include an audio-based automatic speech recognition process and an image-based automatic speech recognition process. AVSR may combine results from both processes with feature fusion. Audio-based speech recognition process may analysis audio according to any method described herein, for instance using a Mel frequency cepstrum coefficients (MFCCs) and/or log-Mel spectrogram derived from raw audio samples. Image-based speech recognition may perform feature recognition to yield an image vector. In some cases, feature recognition may include any feature recognition process described in this disclosure, for example a variant of a convolutional neural network. In some cases, AVSR employs both an audio datum and an image datum to recognize verbal content 1236. For instance, audio vector and image vector may each be concatenated and used to predict speech made by a subject 1220, who is ‘on camera.’

In some cases, computing device 1204 may be configured to recognize at least a keyword 1240 as a function of visual verbal content. In some cases, recognizing at least keyword 1240 may include optical character recognition. As used in this disclosure, a “keyword” is an element of word or syntax used to identify and/or match elements to each other. In some cases, computing device 1204 may transcribe much or even substantially all verbal content 1236 from target resume video 1208.

Still referring to FIG. 12, acquiring plurality of video elements may include identifying a series of frames of target video resume 1208, identifying a corresponding series of frames of at least an existing video resume 1212, and comparing the series of frames of target video resume 1208 to existing video resume 1212. An existing video resume is video resume 1212 that has been uploaded to a video resume database. The series of frames may include a group of pictures having some degree of internal similarity, such as a group of pictures representing a scene. In some embodiments, comparing series of frames may include video compression by inter-frame coding. The “inter” part of the term refers to the use of inter frame prediction. This kind of prediction tries to take advantage from temporal redundancy between neighboring frames enabling higher compression rates. Video data compression is the process of encoding information using fewer bits than the original representation. Any compression is either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by removing

unnecessary or less important information. Typically, a device that performs data compression is referred to as an encoder, and one that performs the reversal of the process (decompression) as a decoder. Compression is useful because it reduces the resources required to store and transmit data. Computational resources are consumed in the compression and decompression processes.

5 Data compression is subject to a space–time complexity trade-off. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in full before watching it may be inconvenient or require additional storage. Video data may be represented as a series of still image frames. Such data usually contains abundant amounts of spatial and
10 temporal redundancy. Video compression algorithms attempt to reduce redundancy and store information more compactly.

Still referring to FIG. 12, inter-frame coding works by comparing each frame in the video with the previous one. Individual frames of a video sequence are compared from one frame to the next, and the video compression codec sends only the differences to the reference frame. If
15 the frame contains areas where nothing has moved, the system can simply issue a short command that copies that part of the previous frame into the next one. If sections of the frame move in a simple manner, the compressor can emit a (slightly longer) command that tells the decompressor to shift, rotate, lighten, or darken the copy. Usually, the encoder will also transmit a residue signal which describes the remaining more subtle differences to the reference imagery. Using
20 entropy coding, these residue signals have a more compact representation than the full signal. In areas of video with more motion, the compression must encode more data to keep up with the larger number of pixels that are changing. As used in this disclosure, reference frames are frames of a compressed video (a complete picture) that are used to define future frames. As such, they are only used in inter-frame compression techniques. Some modern video encoding standards,
25 such as H.264/AVC, allow the use of multiple reference frames. This allows the video encoder to choose among more than one previously decoded frame on which to base each macroblock in the next frame. While the best frame for this purpose is usually the previous frame, the extra reference frames can improve compression efficiency and/or video quality. The two frame types used in inter-frame coding is P-frames and B-frames. A P-frame (Predicted picture) holds only
30 the changes in the image from the previous frame. For example, in a scene where a car moves

across a stationary background, only the car's movements need to be encoded. The encoder does not need to store the unchanging background pixels in the P-frame, thus saving space. A B-frame (Bidirectional predicted picture) saves even more space by using differences between the current frame and both the preceding and following frames to specify its content. An inter coded frame is divided into blocks known as macroblocks. A macroblock is a processing unit in image and video compression formats based on linear block transforms, typically the discrete cosine transform (DCT). A macroblock typically consists of 16×16 samples, and is further subdivided into transform blocks, and may be further subdivided into prediction blocks. Formats which are based on macroblocks include JPEG, where they are called MCU blocks, H.261, MPEG-1 Part 2, H.262/MPEG-2 Part 2, H.263, MPEG-4 Part 2, and H.264/MPEG-4 AVC. After the inter coded frame is divided into macroblocks, instead of directly encoding the raw pixel values for each block, the encoder will try to find a block similar to the one it is encoding on a previously encoded frame, referred to as a reference frame. This process is done by a block matching algorithm. If the encoder succeeds on its search, the block could be encoded by a vector, known as motion vector, which points to the position of the matching block at the reference frame. The process of motion vector determination is called motion estimation. In most cases the encoder will succeed, but the block found is likely not an exact match to the block it is encoding. This is why the encoder will compute the differences between them. Those residual values are known as the prediction error and need to be transformed and sent to the decoder. To sum up, if the encoder succeeds in finding a matching block on a reference frame, it will obtain a motion vector pointing to the matched block and a prediction error. Using both elements, the decoder will be able to recover the raw pixels of the block. For example, video resumes 1212 and 1208 may be compressed using a P-frame algorithm and broken down into macroblocks. Individual still images taken from video resume 1212 can then be compared against a reference frame taken from target video resume 1208. A P-frame from video resume 1212 would only hold the changes in image from target video resume 1208. For example, if both video resumes 1212 and 1208 include a subject 1220 speaking in a similar manner, then what would be encoded and stored would be subtle changes such as a foreign object entering the background of video resume 1212 compared to no foreign object in the background of the reference frame from target video resume 1208. Exemplary video compression codecs include without limitation H.26x codecs, MPEG

formats, VVC, SVT-AV1, and the like. In some cases, compression may be lossy, in which some information may be lost during compression. Alternatively, or additionally, in some cases, compression may be substantially lossless, where substantially no information is lost during compression. In some cases, image component 1216a-b may include a plurality of temporally sequential frames. In some cases, each frame may be encoded (e.g., bitmap or vector-based encoding). Each frame may be configured to be displayed by way of a display. Exemplary displays include without limitation light emitting diode (LED) displays, cathode ray tube (CRT) displays, liquid crystal displays (LCDs), organic LEDs (OLEDs), quantum dot displays, projectors (e.g., scanned light projectors), and the like.

With continued reference to FIG. 12, computing device 1204 is configured to compare at least existing video resume 1212 and target video resume 1208 to obtain a comparison result. An initial pass may be used by computing device 1204 to sort elements of video resumes 1212 into categories, and a subsequent pass may involve detailed comparison of category-matched video elements from at least two video resumes 1212 to one another. For example, the initial pass may include classifying the plurality of existing video resumes 1212 based on, image component 1216a, audio component, attributes 1224, or at least identifying subject 1220. For example, identifying subject 1220 could include name of user or subject 1220, account number, social security number, telephone number, address, and the like. In some embodiments, computing device 1204 may utilize a candidate classifier, which may include any classifier used throughout this disclosure, to run an initial pass over the video elements of video resumes 1212, break down and categorizes such elements before comparing it to target video resume 1208. A “classifier,” as used in this disclosure is a machine-learning model, such as a mathematical model, neural net, or program generated by a machine learning algorithm known as a “classification algorithm that sorts inputs into categories or bins of data, outputting the categories or bins of data and/or labels associated therewith. A classifier may be configured to output at least a datum that labels or otherwise identifies a set of data that are clustered together, found to be close under a distance metric, or the like. As used in this disclosure, a “candidate classifier” is a classifier that classifies subjects 1220 to target video resume 1212 or an overall job description. In some cases, candidate classifier may include a trained machine-learning model, which is trained using candidate training data. As used in this disclosure, “candidate training data” is a training data that

correlates one or more of subjects 1220, subject-specific data, and subject attributes 1224 to one or more job descriptions, description-specific data, and job description data. As used in this disclosure, a “job description datum” is an element of information associated with a job description. Target video resume 1208 may be representative of such job descriptive data. For example, in the initial pass, video resumes 1212 may be categorized based on subject’s 1220 attributes 1224 such as credentials. As used in this disclosure, “credentials” are any piece of information that indicates an individual’s qualification to perform a certain task or job. All existing video resumes 1212 may be grouped based on level of experience, educational history, certifications, and the like.

Still referring to Fig. 12, after the initial pass, during the subsequent pass, two or more existing video resumes 1212 may be compared against one another and ranked based on similarity or relevance to target video resume 1208 before an overall comparison result of video resumes 1212 and target video 1208 is computed. As used in this disclosure, “relevance” is a measure of closeness of association. For example, if candidate classifier grouped video resumes 1212 together based on years of experience, candidate classifier may then compare the video resumes in that group to output a relevance metric of video resumes 1212 that most align with target video 1208. In some cases, relevance metric may be a quantified metric, for example in arbitrary units or relative unit (e.g., percent).

Still referring to FIG. 12, in some embodiments, after the initial and subsequent pass are ran, computing device may utilize the data gathered from the candidate classifier to calculate an overall comparison score of existing video resumes 1212 to target video resume 1208. In some cases, comparison result may contain a comparison score 1248 that represents a degree of similarity between target video resume 1208 and existing video resume 1212 of the plurality of existing video resumes. Comparison score 1248 may be determined by dynamic time warping (DTW) based on a similarity matrix. Dynamic time warping may include algorithms for measuring similarity between two sequences, which may vary in time or speed. For instance, similarities in walking patterns may be detected, even if in one video the person was walking slowly and if in another he or she were walking more quickly, or even if there were accelerations and deceleration during one observation. DTW has been applied to video, audio, and graphics – indeed, any data that can be turned into a linear representation can be analyzed with DTW. In

some cases, DTW may allow computing device 1204 to find an optimal match between two given sequences (e.g., time series) with certain restrictions. That is, in some cases, sequences can be “warped” non-linearly to match each other.

With continued reference to FIG. 12, in some embodiments, computing device 1204 may
5 extract or otherwise recognize at least feature 1228 from video resume 1212. Feature 1228 may be recognized and/or extracted from image component 1216a of existing video resumes 1212. In some cases, features 1228 may be recognized, which are associated with non-verbal content 1232. For example, in some cases, visual non-verbal content such as expression of subject’s 1220 emotion may be represented by a number of features 1228 which are readily extracted from
10 image component 1216a of existing video resumes 1212. In some cases, recognition and/or extraction of features 1228 from image component 1216a may include use of machine vision techniques.

Still referring to FIG. 12, in some embodiments, apparatus 1200 may include a machine vision process. A machine vision process may use image component 1216a from video resume
15 1212, to make a determination about verbal 1236 and/or non-verbal content 1232. For example, in some cases a machine vision process may be used for world modeling or registration of objects within a space. In some cases, registration and/or feature recognition may include image processing, such as without limitation object recognition, feature 1228 detection, edge/corner detection, and the like. Non-limiting example of feature detection 1228 may include scale
20 invariant feature transform (SIFT), Canny edge detection, Shi Tomasi corner detection, and the like. In some cases, a machine vision process may operate image classification and segmentation models, such as without limitation by way of machine vision resource (e.g., OpenMV or TensorFlow Lite). A machine vision process may detect motion, for example by way of frame differencing algorithms. A machine vision process may detect markers, for example blob
25 detection, object detection, face detection, and the like. In some cases, a machine vision process may perform eye tracking (i.e., gaze estimation). In some cases, a machine vision process may perform person detection, for example by way of a trained machine learning model. In some cases, a machine vision process may perform motion detection (e.g., camera motion and/or object motion), for example by way of optical flow detection. In some cases, machine vision

process may perform code (e.g., barcode) detection and decoding. In some cases, a machine vision process may additionally perform image capture and/or video recording.

Still referring to FIG. 12, in some cases, machine vision process may perform pose-estimation for example to ascertain a relative location or movement of objects within existing video resumes to include one or more transformations, for example to a view of a frame (or an image or existing video resumes) relative a three-dimensional coordinate system; exemplary transformations include without limitation homography transforms and affine transforms. In an embodiment, registration of first frame to a coordinate system may be verified and/or corrected using object identification and/or computer vision, as described above. For instance, and without limitation, an initial registration to two dimensions, represented for instance as registration to the x and y coordinates, may be performed using a two-dimensional projection of points in three dimensions onto a first frame, however. A third dimension of registration, representing depth and/or a z axis, may be detected by comparison of two frames; image recognition and/or edge detection software may be used to detect multiple views of images of an object (from subsequent frames) to derive a relative position along a third (z) axis. In some cases, solicitation video may include a stereo image, having two stereoscopic views, which may be compared to derive z-axis values of points on object permitting, for instance, derivation of further z-axis points within and/or around the object using interpolation. Alternatively, or additionally, relative movement within Image component 1216 (e.g., frame to frame) may be used to ascertain positions of objects, even along a z-axis, for instance by way of kinetic parallax. In some cases, relative motion of objects further away may occur at a different speed than objects nearby, this phenomenon may be used to ascertain a position of objects relative a camera, for example when the camera is moving. Object recognition and pose estimation may be repeated with multiple objects in field of view, including without a subject. In an embodiment, x and y axes may be chosen to span a plane common to a field of view of a camera used for solicitation video image capturing and/or an xy plane of a first frame; a result, x and y translational components and ϕ may be pre-populated in translational and rotational matrices, for affine transformation of coordinates of object, also as described above. Initial x and y coordinates and/or guesses at transformational matrices may alternatively or additionally be performed between first frame and second frame, as described above. For each point of a plurality of points on object and/or edge

and/or edges of object as described above, x and y coordinates of a first frame may be populated, with an initial estimate of z coordinates based, for instance, on assumptions about object, such as an assumption that ground is substantially parallel to an xy plane as selected above. Z

coordinates, and/or x, y, and z coordinates, registered using image capturing and/or object

5 identification processes as described above may then be compared to coordinates predicted using initial guess at transformation matrices; an error function may be computed using by comparing the two sets of points, and new x, y, and/or z coordinates, may be iteratively estimated and compared until the error function drops below a threshold level.

Still referring to FIG. 12, in some cases, a machine vision process may use at least an
10 image classifier, or any classifier described throughout this disclosure. As a non-limiting example, a machine vision process may use an image classifier, wherein the input is image component 1216a of video resumes 1212, and through a classification algorithm, outputs image components 1216a into categories based on training data, such as sequential video resume 1212 frames that match target video resume 1208. Computing device 1204 and/or another device may
15 generate a classifier using a classification algorithm, defined as a process whereby a computing device derives a classifier from training data. Classification may be performed using, without limitation, linear classifiers such as without limitation logistic regression and/or I Bayes classifiers, nearest neighbor classifiers such as k-nearest neighbors classifiers, support vector machines, least squares support vector machines, fisher's linear discriminant, quadratic
20 classifiers, decision trees, boosted trees, random forest classifiers, learning vector quantization, and/or neural network-based classifiers.

Still referring to FIG. 12, in some cases, computing device associate subject 1220 with at least feature 1228 from the target video resume 1208 by performing a text retrieval process as a function of at least keyword 1240. For example, in some cases, computing device 1204 may
25 query feature 1228 from target video resume 1208 for presence of at least keyword 1240. Querying feature 1228 from target video resume 1208 may include one or more of word searching, phrase searching, proximity searching, full-text searching, field (or metadata) searching, and the like. As used in this disclosure, a "target video feature keyword" is a word that is of particular relevance to an individual feature of target video resume.

With continued reference to FIG. 12, in some embodiments, querying at least image component 1216a-b may be performed with a text search, for example using at least keyword 1240 as a search term. Text search may include techniques for searching a single computer-stored document or a collection of documents, for example in a database. Text search may include full-text search. Full-text search may be distinguished from searches based on metadata or on field-based searching (e.g., fields such as titles, abstracts, selected sections, or bibliographical references). In an exemplary full-text search, computing device 1204 may examine all words in every stored document as it tries to match search criteria (for example, keywords). Alternatively, a text search may be limited to fields, such as with field-based searching.

Still referring to FIG. 12, in some embodiments, text searching may include querying a database of video resumes in which multiple video resumes 1212 of subjects 1220 are stored. As used in this disclosure, “video resume database” is a data structure configured to store data associated with a plurality of video resumes [...]. Database may be implemented, without limitation, as a relational database, a key-value retrieval database such as a NOSQL database, or any other format or structure for use as a database that a person skilled in the art would recognize as suitable upon review of the entirety of this disclosure. Database may alternatively or additionally be implemented using a distributed data storage protocol and/or data structure, such as a distributed hash table or the like. Database may include a plurality of data entries and/or records as described above. Data entries in a database may be flagged with or linked to one or more additional elements of information, which may be reflected in data entry cells and/or in linked tables such as tables related by one or more indices in a relational database. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which data entries in a database may store, retrieve, organize, and/or reflect data and/or records as used herein, as well as categories and/or populations of data consistently with this disclosure. In some cases, querying of at least a video element may include any number of querying tools, including without limitation keywords (as described above), field-restricted search, Boolean queries, phrase search, concept search, concordance search, proximity search, regular expression, fuzzy search, wildcard search, and the like. In some cases, keywords may be used to perform a query. In some cases, a document (or trained indexers) may supply a list of words that describe

subject of the document, including without limitation synonyms of words that describe the subject. In some cases, keywords may improve recall, for instance if the keyword list includes a keyword that is not in text of a document. In some cases, querying tools may include field-restricted search. A field-restricted search may allow a queries scope to be limited to within a particular field within a stored data record, such as “Title” or “Author.” In some cases, a query tool may include Boolean queries. Searches that use Boolean operators (for example, “encyclopedia” AND “online” NOT “Encarta”) can dramatically increase precision of a search. In some cases, an AND operator may say, in effect, “Do not retrieve any document unless it contains both of these terms.” In some cases, a NOT operator may say, in effect, “Do not retrieve any document that contains this word.” In some cases, a retrieval list retrieving too few documents, may prompt and OR operator to be used in place of an AND operator to increase recall; consider, for example, “encyclopedia” AND “online” OR “Internet” NOT “Encarta”. This search will retrieve documents about online encyclopedias that use the term “Internet” instead of “online.” In some cases, search precision and recall are interdependent and negatively correlated in text searching. In some cases, a query tool may include phrase search. In some cases, a phrase search may match only those documents that contain a specified phrase. In some cases, a query tool may include a concept search. In some cases, a concept search may be based on multi-word concepts, for example compound term processing. In some cases, a query tool may include a concordance search. In some cases, a concordance search may produce an alphabetical list of all principal words that occur in a text and may include their immediate context. In some cases, a query tool may include a proximity search. In some cases, a proximity search matches only those documents that contain two or more words that are separated by a specified number of words, are in the same sentence, or an in the same paragraph. A query tool may include a regular expression. In some cases, a regular expression may employ a complex but powerful querying syntax that can be used to specify retrieval conditions with precision, for instance database syntax. A query tool may include a fuzzy search. In some cases, a fuzzy search may search for a document that matches given terms while allowing for some variation around them. In some cases, a query tool may include a wildcard search. In some cases, a wildcard search may substitute one or more characters in a search query for a wildcard character such as an asterisk.

For example, using a wildcard, such as an asterisk, in a search query “s*n” will search for terms inclusive of “sin,” “son,” “sun,” and the like.

Still referring to FIG. 12, in some cases association of existing video resume 1212 with at least target video resume 1208 may include one or more of machine-learning process 1244
5 and/or classifiers 1252. Machine-learning process 1244 may include any machine-learning process described in this disclosure. Classifier 1252 may include any classifier described in this disclosure. In some cases, computing device 1204 may associate existing video resume 1212 with at least keyword 1240 as a function of at least feature 1228. As described above, at least feature 1228 may represent non-verbal content 1232 from existing video resume 1212, at least
10 feature 1228 may be correlated to at least target video resume 1208, for example by way of one or more models (e.g., machine-learning models). In some cases, computing device 1204 associate video resume 1212 with at least target video resume 1208 by correlating, matching, or otherwise comparing subject-specific data with description-specific data. In some cases, one or both of subject-specific data and description-specific data may be at least partially represented by
15 features 1228 which are inputs and/or outputs of machine-learning processes. For example, a machine-learning model may be trained with deterministic or historical subject-specific data correlated to description-specific data.

Still referring to FIG. 12, as a function of the comparison result, a ranking of the plurality of existing resume videos may be generated. In some cases, generating the ranking of the
20 plurality of existing resumes may include linear regression techniques. Computing device 1204 may be designed and configured to create a machine-learning model using techniques for development of linear regression models. Linear regression models may include ordinary least squares regression, which aims to minimize the square of the difference between predicted outcomes and actual outcomes according to an appropriate norm for measuring such a difference
25 (e.g., a vector-space distance norm); coefficients of the resulting linear equation may be modified to improve minimization. Linear regression models may include ridge regression methods, where the function to be minimized includes the least-squares function plus term multiplying the square of each coefficient by a scalar amount to penalize large coefficients. Linear regression models may include least absolute shrinkage and selection operator (LASSO) models, in which ridge
30 regression is combined with multiplying the least-squares term by a factor of 12 divided by

double the number of samples. Linear regression models may include a multi-task lasso model wherein the norm applied in the least-squares term of the lasso model is the Frobenius norm amounting to the square root of the sum of squares of all terms. Linear regression models may include the elastic net model, a multi-task elastic net model, a least angle regression model, a
5 LARS lasso model, an orthogonal matching pursuit model, a Bayesian regression model, a logistic regression model, a stochastic gradient descent model, a perceptron model, a passive aggressive algorithm, a robustness regression model, a Huber regression model, or any other suitable model that may occur to persons skilled in the art upon reviewing the entirety of this disclosure. Linear regression models may be generalized in an embodiment to polynomial
10 regression models, whereby a polynomial equation (e.g., a quadratic, cubic or higher-order equation) providing a best predicted output/actual output fit is sought; similar methods to those described above may be applied to minimize error functions, as will be apparent to persons skilled in the art upon reviewing the entirety of this disclosure.

With continued reference to FIG. 12, computing device 1204 may be configured to
15 determine, as a function of the comparison result, a duplication coefficient 1256 for target resume video 1208. As used in this disclosure, “duplication coefficient” is a quantitative value of observed similarities between two or more video resumes. Duplication coefficient 1256 may be calculated or computed to provide a measure or metric of similarity between target video resume 1208 and at least existing resume video 1212. Duplication coefficient 1256 could stand for “how
20 much”, “how many” or “how often” data appears in video resumes 1212. The different categories of quantitative data could include, measurements, counts, calculations, sensors, projections, quantification of qualitative data, and the like. For example, duplication coefficient 1256 may be the average of the observed subject-specific data duplications of existing video resumes, divided by the total number of existing video resumes 1212 in a database. Other
25 examples, duplication coefficient could measure the number of subjects 1220 appearing per video resume 1212. Duplication coefficient 1256 could quantify how many subjects 1220 answered an interview question. Duplication coefficient could count subjects 1220 having similar technical backgrounds to target video resume 1208. Another example, duplication coefficient 1256 could project subjects 1220 most qualified for a position based on their video
30 resumes. In some embodiments, duplication coefficient 1256 may include a metric range on a

scale of 0, where 0 is least like target video resume 1208, to 120, where 120 is exact same as target video 1208. It could also include a range of percentages and may cover any suitable range or rating score. In some cases, determining duplication coefficient 1256 for target resume video 1208 may include linear regression analysis, machine-learning process, and the like. For
5 example, duplication coefficient 1256 may be calculated by using classifier 1252 configured to output at least a datum that labels or otherwise identifies a set of data that are clustered together, found to be close under a distance metric. Additionally, or alternatively, duplication coefficient 1256 may be an output of a machine-learning module 1244 that generates classifier 1252 using a classification algorithm.

10 Still referring to FIG. 12, computing device may utilize duplication coefficient 1256 in determining which video resumes have value for storage purposes. Duplication coefficient 1256 may be used to measure the quantity of video resumes matching video elements of target video resume 1208. Duplication coefficient 1256 may be used to select subject-specific data among existing video resumes 1212 and match the elements therein to give a range of similarity among
15 all videos. For example, duplication coefficient 1256 may be the metric range of similarity between existing video resumes 1212 based on image component 1216a, audio component, non-verbal content 1232, or verbal content 1236. Computing device 1204 may take duplication coefficient of video resumes 1212 image components 1216a and compare to image component 1216b of target video resume and give a percentage or other numeric value illustrating video
20 resumes 1212 image components that are most similar to target video resume 1208 image component 1216b. Computing device 1204 may utilize the duplication coefficient to disregard, delete, flag, video resumes 1212 based on a threshold requirement set by the user or subject 1220. For example, a user may set computing device 1204 to disregard all video resumes with attributes 1224 dissimilar to target video resume 1208. A user could set computing 1204 to delete
25 video resumes 1212 that that have an audio competent less than 50% similar to target video resume 1208 audio component. Computing device 1204 may be set to flag video resumes 1212 that are missing and image or audio component in the video file.

Still referring to FIG. 12, computing device 1204 may utilize the duplication coefficient to output indications regarding video resume to a user, such as without limitations indications
30 identifying flagged, matching, nonrelevant, and/or duplicative video resumes 1212. Indications

may alternatively or additionally display a degree of similarity to another video and/or video resume, may list one or more video resumes found to be similar as a function of duplication coefficient, or the like. In some cases, computing device 1204 may utilize the duplication coefficient to automatically reject video resumes, or lower overall subject's 1220 score, or other scores that measures subject's 1220 desirability as an employee, for instance and without limitation based on a threshold requirement of target video resume 1208. For example, duplication coefficient 1256 may be part of a scoring system that generates a score by an algorithm based on user data records. Scoring system may include, without limitation, a career scoring system producing dynamic rating that represent, for instance in the form of a numerical score, a degree of employability, suitability for a job, and the like. Additional disclosure related to the scoring system of subject 1220 may be found in U.S. Patent Application No. 17/486,461 entitled "SYSTEMS AND METHODS FOR SCORE GENERATION FOR APPLICANT TRACKING CROSS-REFERENCE TO RELATED APPLICATIONS" by W. Inman et al., which is incorporated in its entirety herein by reference.

Referring now to FIG. 13, an exemplary frame of a video resume 1300 is illustrated. Frame may include at least a portion of an image component. In some cases, a video resume may include many frames, for instance at a frame rate (e.g., 24, 30, 60, or 120 frames per second), thereby facilitating video.

With continued reference to FIG. 13, frame may include an image of subject. Alternatively, or additionally, in some cases video resume may include images of other people or no person at all. An image of subject may be included in video resume.

Still referring to FIG. 13, as more potential employees become more literate with video communication, video communication within a work environment may gain in popularity. Sharing one's accomplishments and advertising one's services represents an important application of communication for an individual jobseeker. As jobseekers gain in ability and confidence in video communication, they will choose to put their best foot forward using the medium they believe best facilitates advantageous communication. As a result, some embodiments of the present disclosure inventively anticipate that video resumes will gain in popularity and address an unmet need associated with video resumes. Presently, employers can screen many written resumes automatically. However, this technology cannot be used with video

resumes. In some case, embodiments described herein improve upon present resume screening systems by allowing the automatic parsing and correlating of video resumes.

Referring now to FIG. 14, a table 1400 is depicted that illustrates an association between target video resume 1404 and plurality of existing video resumes 1408a-d. Table 1400 illustrates an image component of target video resume 1404 being associated with plurality of existing video resumes 1408a-d.

With continued reference to FIG. 14, table 1400 may include a few columns, for example a ranking column, video resume 1408a-d column, and a factor (e.g., comparison result 1416) column. Table 1400 may include more than one factor column corresponding to other factors. For example, relevance, audio, text, video frame rate, and the like. Ranking 1412 may include a ranking of video resumes 1408a-d, for example from most similar to target resume video resume 1404, video resume 1408a, to a least similar, video resume 1408d.

Referring now to FIG. 15, an exemplary video resume database 1500 is illustrated by way of a block diagram. Computing device 1204 may be communicatively connected with video resume database 1500. For example, in some cases, video resume database 1500 may be local to computing device 1204. Alternatively, or additionally, in some cases, video resume database 1500 may be remote to computing device 1204 and communicative with computing device 1204 by way of one or more networks. As used in this disclosure, “video resume database” is a data structure configured to store data associated with a plurality of video resumes 1504. In some cases, data within video resume database 1500 may be labeled to indicate which video resume 1504 is associated with the data, for instance with a unique video resume identification number. Video resume database 1500 may store a plurality of video resumes 1504. Video resumes 1504 may be uploaded to video resume database 1500 from at least a remote device. Remote device may include any computing device described in this disclosure. As described above with reference to FIG. 12, video resumes 1504 may include image components 1508, non-verbal contents 1512, and/or verbal contents 1516. Additionally, video resume database 1500 may include video resume keywords 1520. As described above, video resume keywords 1520 may be include words that were represented verbally within video resume 1504. Alternatively, or additionally, video resume keywords 1520 may be associated and/or classified to video resume 1504 from non-verbal content 1512 or through user entry. Video resume database 1500 may

additionally store video resume features 1524 associated with video resumes 1504. As described above, video resume features 1524 may be recognized or extracted from video resumes 1504 associated with a subject by way of a number of processes described in detail in this disclosure.

Referring now to FIG. 16, a training data database 1600 is illustrated by way of a block diagram. Computing device 1204 may be communicatively connected with video resume database 1500. For example, in some cases, training data database 1600 may be local to computing device 1204. Alternatively, or additionally, in some cases, training data database 1600 may be remote to computing device 1204 and communicative with computing device 1204 by way of one or more networks. As used in this disclosure, “training data database” is a data structure configured to store training data. As described in this disclosure, many embodiments may use training data to perform any number of processes. Training data may, in some cases, need to be used, re-used, modified, deleted, and the like. In some cases, training data may need to be conditionally selected. Additionally, training data sets may be stored in training data database 1600. For example, in some cases, training data that may be used to extract data (e.g., keywords and/or features), for example from video resumes may be stored in training data database 1600; these training sets may include without limitation visual verbal content training data 1604, visual non-verbal content training data 1608.

[0001] Referring now to FIG. 17, an exemplary method of parsing and comparing resume video duplications is illustrated by way of a flow diagram. At step 1705, method may include acquiring, using a computing device, a plurality of video elements from a target video resume. This may be accomplished without limitation as described above in reference to FIGS. 1-16, wherein target video includes at least an image component. In some cases, image component may include non-verbal or verbal content, for example with reference to FIG. 12. Computing device may include any computing device as described in this disclosure. Target video resume may include any target video resume described in this disclosure, for example with reference to FIGS. 1-66. Subject may include any subject described in this disclosure, for example with reference to FIG. 12.

[0002] With continued reference to FIG. 17, at step 1710, method may include comparing, using a computing device, at least an existing video resume and the target video resume video resume for similarities. Existing video resume may be any video resume as described above in

this disclosure. In some cases, computing device may obtain a comparison result of video resume and target video resume. The comparison result may contain a comparison score that represents a degree of similarity between target video resume and an existing video resume of the plurality of existing video resumes, as described in this disclosure. In some cases, the comparison score may
5 be determined by dynamic time warping based on a similarity matrix, as described in this disclosure. As a function of the comparison result, a ranking of the plurality of existing resume videos is may be generated. In some cases, generating the ranking of the plurality of existing resumes may include linear regression techniques, as disclosed in FIGS. 12.

[0003] With continued reference to FIG. 17, at step 1715 method may include determining,
10 using computing device, as a function of the comparison result, a duplication coefficient for target resume video, wherein the duplication coefficient relates to a similarity between target video resume and at least an existing resume video, as disclosed in FIG. 12. Duplication coefficient may represent any quantitative value as disclosed in FIG. 12 and may be calculated through a computing device or machine-learning process as disclosed in FIG. 12.

[0004] With continued reference to FIG. 17, in some embodiments, method 700 may include associating, using computing device, video resume with target video resume as a function of at least a keyword, as described with reference to FIG. 12. In some embodiments, method may additionally include classifying, using computing device and at least a candidate classifier, video resume to the target video resume. Candidate classifier may include any
20 classifier described in this disclosure, for example including with reference to FIG. 12. In some embodiments, method 1700 may additionally include querying, using computing device, a video resume with at least a keyword. In some embodiments, method 1700 may additionally include determining relevance as a function of association between video resume and target video resume.

[0005] Still referring to FIG. 17, in some embodiments, at least an image component may include visual non-verbal content related to at least an attribute of subject. Visual non-verbal content may include any visual non-verbal content described in this disclosure, for example including with reference to FIG. 12. Attribute may include any attribute described in this disclosure, for example including with reference to FIG. 12. In some cases, method 1700 may
30 additionally include recognizing, using computing device, at least a feature as a function of

visual non-verbal content and associating, using the computing device, subject with target video resume as a function of the at least a feature. Feature may include any feature described in this disclosure, for example including with reference to FIG. 12. In some cases, recognizing at least a feature and associating video resume with target video resume as a function of the at least a feature may use a machine-learning process. Machine-learning process may include any machine learning process described in this disclosure.

[0006] Still referring to FIG. 17, in some embodiments at least an image component may include visual verbal content. In some cases, method 1700 may additionally include recognizing, using computing device, at least a keyword as a function of visual verbal content and associating, using the computing device, video resume with target video resume as a function of the at least a keyword. Visual verbal content may include any visual verbal content described in this disclosure, for example including with reference to FIG. 12.

[0007] Still referring to FIG. 17, in some embodiments at least an audio component may include audible non-verbal content related to at least an attribute of subject. Audible non-verbal content may include any audible non-verbal content described in this disclosure, for example including with reference to FIG. 12. In some cases, method 1700 may additionally include recognizing, using computing device, at least a feature as a function of audible non-verbal content and associating, using the computing device, video resume with target video resume as a function of the at least a feature.

Referring now to FIG. 18, an exemplary embodiment of a digital career development apparatus 1800 (also referred to in this disclosure as a “career coaching apparatus” or “apparatus”) is illustrated in accordance with one or more embodiments of the present disclosure. In one or more embodiments, apparatus 1800 includes a computing device 1804. Computing device 1804 may include any computing device 1804 as described in this disclosure, including without limitation a processor (*e.g.*, processor 1844), control circuit, microcontroller, microprocessor, digital signal processor (DSP), and/or system on a chip (SoC). Computing device may include a computer system with one or more processors (*e.g.*, CPUs), a graphics processing unit (GPU), or any combination thereof. Computing device may include a memory component, such as memory component 1840, which may include a memory, such as a main memory and/or a static memory, as discussed further in this disclosure below. Computing device

may include a display component, as discussed further below in the disclosure. In one or more embodiments, computing device 1804 may include, be included in, and/or communicate with a mobile device, such as a mobile telephone, smartphone, tablet, and the like. Computing device 1804 may include a single computing device operating independently, or may include two or more computing devices operating in concert, in parallel, sequentially, or the like. Two or more computing devices may be included together in a single computing device or in two or more computing devices. Computing device 1804 may interface or communicate with one or more additional devices, as described below in further detail, via a network interface device. Network interface device may be utilized for connecting computing device 1804 to one or more of a variety of networks, and one or more devices. Examples of a network interface device include, but are not limited to, a network interface card (*e.g.*, a mobile network interface card, a LAN card), a modem, any combination thereof, and the like. Examples of a network include, but are not limited to, a wide area network (*e.g.*, the Internet, an enterprise network), a local area network (*e.g.*, a network associated with an office, a building, a campus or other relatively small geographic space), a telephone network, a data network associated with a telephone/voice provider (*e.g.*, a mobile communications provider data and/or voice network), a direct connection between two computing devices, and any combinations thereof. A network may employ a wired and/or a wireless mode of communication. In general, any network topology may be used. Information (*e.g.*, data, software etc.) may be communicated to and/or from a computer and/or a computing device. Computing device 1804 may include but is not limited to, for example, a computing device or cluster of computing devices in a first location and a second computing device or cluster of computing devices in a second location. Computing device 1804 may include one or more computing devices dedicated to data storage, security, distribution of traffic for load balancing, and the like. Computing device 1804 may distribute one or more computing tasks, as described below, across a plurality of computing devices of computing device 1804, which may operate in parallel, in series, redundantly, or in any other manner used for distribution of tasks or memory between computing devices. Computing device 1804 may be implemented using a “shared nothing” architecture in which data is cached at the worker, in an embodiment, this may enable scalability of apparatus 1800 and/or computing device 1804.

With continued reference to FIG. 18, computing device 1804 may be designed and/or configured to perform any method, method step, or sequence of method steps in any embodiment described in this disclosure, in any order and with any degree of repetition. For instance, computing device 1804 may be configured to perform a single step or a sequence of steps
5 repeatedly until a desired outcome or commanded outcome is achieved. Repetition of a step or a sequence of steps may be performed iteratively and/or recursively using outputs of previous repetitions as inputs to subsequent repetitions, aggregating inputs, and/or outputs of repetitions to produce an aggregate result, reduction or decrement of one or more variables such as global variables, and/or division of a larger processing task into a set of iteratively addressed smaller
10 processing tasks. Computing device 1804 may perform any step or sequence of steps as described in this disclosure in parallel, such as simultaneously and/or substantially simultaneously performing a step two or more times using two or more parallel threads, processor cores, or the like. Division of tasks between parallel threads and/or processes may be performed according to any protocol suitable for division of tasks between iterations. Persons
15 skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which steps, sequences of steps, processing tasks, and/or data may be subdivided, shared, or otherwise dealt with using iteration, recursion, and/or parallel processing.

With continued reference to FIG. 18, computing device 1804 is configured to receive a posting datum 1812. For the purpose of this disclosure, “posting datum” is information related to
20 an available and/or open job position. For the purposes of this disclosure, a “job position” (also referred to in this disclosure as a “job”) is a paid occupation with designated tasks associated therewith. A job position may include an employment with an employer, such as work as an employee (part-time or full-time), worker, contractor, self-employed, and the like. For example, and without limitation, posting datum 1812 may include information and/or data from a job
25 posting and/or listing that describes an open job position. Posting datum 1812 may include a job position title, qualifications and/or requirements for the job position, expected responsibilities associated with the job position, benefits with the job position, compensation, geographical location, employer information, and the like. Posting datum 1812 may include information related to an employer’s expectations of a person hired for such a job position. For instance, and
30 without limitations, posting datum 1812 may include minimum qualifications that a candidate

must possess to adequately perform the job position. Qualifications for job position may include education, certification, experience, desired skills and/or abilities, personal qualities, and the like. Posting datum 1812 may also include information that a person hired for the job position may expect from the job position. For instance, and without limitation, posting datum 1812 may
5 include working hours for the job position, a type of salary, degree of professionalism, and the like. In one or more embodiments, posting datum 1812 may include a datum or a plurality of data related to an available job.

In one or more embodiments, posting datum 1812 may be provided to or received by computing device 1804 using various means. In one or more embodiments, posting datum 1812
10 may be provided to computing device 1804 by a user, such as a jobseeker or potential job candidate that is interested in being a candidate or considered for a job position by the employer of the job position. A user may manually input posting datum 1812 into computing device using, for example, a graphic user interface and/or an input device. For example, and without limitation, a user may use a peripheral input device to navigate graphic user interface and provide posting
15 datum 1812 to computing device 1804. Non-limiting exemplary input devices include keyboards, joy sticks, light pens, tracker balls, scanners, tablet, microphones, mouses, switches, buttons, sliders, touchscreens, and the like. In other embodiments, posting datum 1812 may be provided to computing device 1804 by a database over a network from, for example, a network-based platform. Posting datum 1812 may be stored in a database and communicated to computing
20 device 1804 upon a retrieval request from a user and/or from computing device 1804. In other embodiments, posting datum 1812 may be communicated from a third-party application, such as from a third-party application on a third-party server, using a network. For example, posting datum 1812 may be downloaded from a hosting website for job listings. In one or more embodiments, computing device 1804 may extract posting datum 1812 from an accumulation of
25 information provided by a database. For instance, and without limitation, computing device may extract needed information from database regarding the job position and avoid taking any information determined to be unnecessary. This may be performed by computing device 1804 using a machine-learning model, which is described in this disclosure further below.

In one or more embodiments, database 1823 may include inputted or calculated
30 information and datum related to job position and user. A datum history may be stored in a

database 1832. Datum history may include real-time and/or previous inputted posting datum 1812 and user datum 1808. In one or more embodiments, database 1832 may include real-time or previously determined record recommendations and/or previously provided interaction preparations. Computing device 1804 may be communicatively connected with past posting

5 database 1832. For example, and without limitation, in some cases, database 1832 may be local to computing device 1804. In another example, and without limitation, database 1832 may be remote to computing device 1804 and communicative with computing device 1804 by way of one or more networks. A network may include, but is not limited to, a cloud network, a mesh network, and the like. By way of example, a “cloud-based” system can refer to a system which

10 includes software and/or data which is stored, managed, and/or processed on a network of remote servers hosted in the “cloud,” e.g., via the Internet, rather than on local servers or personal computers. A “mesh network” as used in this disclosure is a local network topology in which the infrastructure computing device 1804 connect directly, dynamically, and non-hierarchically to as many other computing devices as possible. A “network topology” as used in this disclosure is an

15 arrangement of elements of a communication network. Network may use an immutable sequential listing to securely store database 1832. An “immutable sequential listing,” as used in this disclosure, is a data structure that places data entries in a fixed sequential arrangement, such as a temporal sequence of entries and/or blocks thereof, where the sequential arrangement, once established, cannot be altered or reordered. An immutable sequential listing may be, include

20 and/or implement an immutable ledger, where data entries that have been posted to the immutable sequential listing cannot be altered.

Database 1832 may include keywords. As used in this disclosure, a “keyword” is an element of word or syntax used to identify and/or match elements to each other. For example, without limitation, a keyword 208 may be “mechanical engineer” in the instance that a job

25 posting is looking for a mechanical engineer to fill a job position. In another non-limiting example, a keyword may be “remote” in an example where the job posting is a remote job. Database 1832 may be implemented, without limitation, as a relational database, a key-value retrieval datastore such as a NOSQL database, or any other format or structure for use as a datastore that a person skilled in the art would recognize as suitable upon review of the entirety

30 of this disclosure.

With continued reference to FIG. 18, computing device 1804 is configured to determine a record recommendation 1816 as a function of posting datum 1812, user datum 1808, and a plurality of records. For the purposes of this disclosure, a “record recommendation” is suggested information for generating or altering a resume specific to a posting, such as a posting for a particular job position. In one or more embodiments, computing device 1804 may process a request submitted, for example, by user to determine record recommendation 1816 once posting datum 1812 has been inputted and received by computing device 1804 or once posting datum 1812 has been retrieved from a database. In one or more embodiments, record recommendation 1816 may include recommended edits or changes to a user’s resume, as discussed further in this disclosure below. In other embodiments, record recommendation 1816 may include an automatedly generated record, as discussed further in this disclosure below. In one or more embodiments, record recommendation 1816 may also include a specialize template in which the template is customized for the specific job position based on the posting datum 1812 to allow a user to readily and easily input their personal information into the template to create their own personalized record, such as a resume.

With continued reference to FIG. 18, computing device 1804 is further configured to receive a user datum 1808, as previously mentioned. For the purposes of this disclosure, “user datum” is personal user information and/or attributes relevant to a job position of a posting. User datum 1808 may be audio and/or visual information related to the user’s personal information, attributes, and/or credentials. For example, user datum may be a video, audio file, text, and the like. User datum may include a user’s prior record, such as a draft resume, personal address, social security number, phone number, employment history, experience level, education, certification, acquired skills, geographical location, expected compensation, job performance acknowledgements (*e.g.*, awards, honors, distinguishments), photograph of user, sample work product, and the like. User datum 1808 may be received by computing device 1804 by the same or similar means described above. For example, and without limitation, user datum 1808 may be provided by a user directly, database, third-party application, remote device, immutable sequential listing, and the like. In non-limiting embodiments, user datum 1808 may be provided as independent or unorganized facts, such as answers to prompted questions provided by computing device 1804 and/or as dependent or organized facts, such as a previously prepared

record that the user made in advance. In one or more embodiments, after receiving posting datum 1812 and user datum 1808, computing device 1804 may determine record recommendation 1816 as a function of posting datum 1812 and user datum 1808. For instance, and without limitation, record recommendation 1816 may include a suggested alteration and/or change, such as an addition or deletion of a portion of previously prepared record. In another instance, and without limitation, record recommendation 1816 may include an automatedly generated record created by computing device 1804. In another instance, and without limitation, record recommendation 1816 may include instructions and/or directions to user describing a process for creating a new customized record, such as a customized resume for a particular job position of a posting. In one or more embodiments, language processing, such as by processor 1840, may be used to identify user-related data from a sample resume and replace the user-related data with user-specific data for user, such as user datum and/or posting datum. In one or more embodiments, computing device may present record recommendation 1816 to a user, such as suggest an addition or deletion of a word or phrase from a previously prepared record, or may automatedly execute record recommendation 1816, such as an automated addition or deletion of a word from a previously prepared record or automatically generate a customized record that computing device 1804 solely creates. Record recommendation 1816 may be presented using, for example and without limitations, using a display of apparatus 1800, as discussed further in this disclosure below.

In one or more embodiments, record recommendation 1816 may include suggested recommendations for a video resume. For instance, and without limitation, computing device 1804 may be configured to compare at least an existing video resume, such as a draft video resume of a user, and target video resume, such as a sample, video resume, to obtain a comparison result. An initial pass may be used by computing device 1804 to sort elements of video resumes into categories, and a subsequent pass may involve detailed comparison of category-matched video elements from at least two video resumes to one another. For example, the initial pass may include classifying the plurality of existing video resumes based on an image component, an audio component, user datum, or at least identifying user indica. For example, identifying indica could include personal information of user such as a name of user or subject, account number, social security number, telephone number, address, and the like. Comparison

result may contain a comparison score that represents a degree of similarity between target video resume and existing video resume of the plurality of existing video resumes. In some embodiments, computing device 1804 may utilize a candidate classifier, which may include any classifier used throughout this disclosure, to run an initial pass over the video elements of video resumes, break down and categorizes such elements before comparing it to target video resume. A “classifier,” as used in this disclosure is a machine-learning model, such as a mathematical model, neural net, or program generated by a machine learning algorithm known as a “classification algorithm that sorts inputs into categories or bins of data, outputting the categories or bins of data and/or labels associated therewith. A classifier may be configured to output at least a datum that labels or otherwise identifies a set of data that are clustered together, found to be close under a distance metric, or the like. As used in this disclosure, a “candidate classifier” is a classifier that classifies users to a target resume or a job position description. In some cases, candidate classifier may include a trained machine-learning model, which is trained using candidate training data. As used in this disclosure, “candidate training data” is a training data that correlates one or more of users and user datum to one or more job descriptions, description-specific data, and posting data. A target video resume may be representative of generic information related to posting data. For example, in the initial pass, video resume may be categorized based on user datum, such as attributes or credentials of user. As used in this disclosure, “credentials” are any piece of information that indicates an individual’s qualification to perform a certain task or job.

In one or more embodiments, computing device 1804 may implement a compatibility algorithm or generate a compatibility machine-learning module, such as machine-learning module 1824, to determine a compatibility score 1836 between user and job position. For the purposes of this disclosure, a “compatibility score” is a measurable value representing a relevancy of a user’s characteristics with qualifications of a job position. In one or more non-limiting embodiments, compatibility score 1836 may be a quantitative characteristic, such as a numerical value within a set range. For example, a compatibility score may be a “2” for a set range of 18-10, where “1” represents a job position and user having a minimum compatibility and “10” represents job position and user having a maximum compatibility. In other non-limiting embodiments, compatibility score 1836 may be a quality characteristic, such as a color coding,

where each color is associated with a level of compatibility. In one or more embodiments, if a compatibility score 1836 is “low”, then a user and a job position are considered to have a minimum compatibility; if a compatibility score 1836 is “high”, then a user and a job position are considered to have a maximum compatibility.

5 For the purposes of this disclosure, a “compatibility algorithm” is an algorithm that determines the relevancy of a user’s characteristics with qualifications of a job position. If user is considering applying to a plurality of job positions, then the compatibility scores between each job position of the plurality of job positions and the user may be ranked so that the user may determine which job position the user is most compatible with of the job positions. Compatibility
10 algorithm may include machine-learning processes that are used to calculate a set of compatibility scores. Machine-learning process may be trained by using training data associated with past calculations and/or information for the job position and user, such as data related to past prior compatibility scores, user datum 1808, user datum history, posting datum 1812, posting datum history, or any other training data described in this disclosure. Compatibility score
15 1836 may be determined by, for example, if a certain numerical value of employment position data matches user data, where the more employment position data that matches user data, the higher the score and the greater the compatibility between the user and the job position. For example, and without limitation, posting datum 1812 may include a qualification of requiring a teacher with at least five years of work experience, and user datum 1808 may include seven
20 years of work experience in teaching, then a numerical value representing compatibility score 1836 may be increased due to the data correlating, thus indicating user is more compatible for the job position because of the provided user datum 1808. In an embodiment, compatibility algorithm may be received from a remote device. In some embodiments, compatibility algorithm is generated by computing device 1804. In one or more embodiments, compatibility algorithm
25 may be generated as a function of a user input.

In one or more embodiments, a machine-learning process may be used to determine compatibility algorithm or to generate a machine-learning model that may directly calculate compatibility score 1836. In one or more embodiments, a machine-learning model may be generated using training data. Training data may include inputs and corresponding predetermined
30 outputs so that a machine-learning module may use the correlations between the provided

exemplary inputs and outputs to develop an algorithm and/or relationship that then allows the machine-learning module to determine its own outputs for inputs. Training data may contain correlations that a machine-learning process may use to model relationships between two or more categories of data elements. The exemplary inputs and outputs may come from a database, such as any database described in this disclosure, or be provided by a user. In other embodiments, a machine-learning module may obtain a training set by querying a communicatively connected database that includes past inputs and outputs. Training data may include inputs from various types of databases, resources, and/or user inputs and outputs correlated to each of those inputs so that a machine-learning module may determine an output, such as compatibility score 1836, for an input, such as posting datum 1812 and user datum 1808. Correlations may indicate causative and/or predictive links between data, which may be modeled as relationships, such as mathematical relationships, by machine-learning processes, as described in further detail below. In one or more embodiments, training data may be formatted and/or organized by categories of data elements by, for example, associating data elements with one or more descriptors corresponding to categories of data elements. As a non-limiting example, training data may include data entered in standardized forms by persons or processes, such that entry of a given data element in a given field in a form may be mapped to one or more descriptors of categories. Elements in training data may be linked to descriptors of categories by tags, tokens, or other data elements.

In one or more embodiments, record recommendation 1816 may include information from posting datum 1812 and user datum 1808 to insert into a new customized record. Record recommendation 1816 may include a formatted record. For example, a format of a resume may vary depending on the field of the job position. For example, and without limitation, categorization, syntax, grammar, section titles, headers, font, margins, and the like may vary depending on the field or area of the job position, such as in the field of engineering, advertisement, medicine, acting, and the like. Computing device 1804 may determine the proper formatting for a customized record or record suggestion using posting datum 1812 and/or or a database that includes generalize resume information and etiquette. Record recommendation 1816 may provide a new customized resume in the format related to the job position associated with the posting datum 1812. In one or more embodiments, record recommendation 1816 may

include a video component, audio components, text components, and combination thereof, and the like. For instance, and without limitation, customized record may include a video resume. As used in this disclosure, a “video resume” is a video in visual and/or audio form to provide a recording promoting a jobseeker for employment, such as for a particular job position. In some cases, video resume may include content that is representative or communicative of an at least attribute of a subject, such as a user. As used in this disclosure, a “subject” is a person such as, for example a jobseeker. Subject user may be represented directly by a video resume. For example, in some cases, an image component of a video resume may include an image of a subject. As used in this disclosure, an “image component” may be a visual representation of information, such as a plurality of temporally sequential frames and/or pictures, related to video resume and target video resume. For example, image component may include animations, still imagery, recorded video, and the like. Attributes may include subject’s skills, competencies, credentials, talents, and the like. In some cases, attributes may be explicitly conveyed within video resume. Alternatively, or additionally, in some cases, attributes may be conveyed implicitly with video resume. Video resume may include a digital video. Digital video may be compressed to optimize speed and/or cost of transmission of video. Videos may be compressed according to a video compression coding format (i.e., codec). Exemplary video compression codecs include H.26x codecs, MPEG formats, VVC, SVT-AV1, and the like. In some cases, compression of a digital video may be lossy, in which some information may be lost during compression. Alternatively, or additionally, in some cases, compression of a digital video may be substantially lossless, where substantially no information is lost during compression.

In some cases, computing device 1804 may include audiovisual speech recognition (AVSR) processes to recognize verbal content in a video resume. For example, computing device 1804 may use image content to aid in recognition of audible verbal content such as viewing user move their lips to speak on video to process the audio content of video resume. AVSR may use image component to aid the overall translation of the audio verbal content of video resumes. In some embodiments, AVSR may include techniques employing image processing capabilities in lip reading to aid speech recognition processes. In some cases, AVSR may be used to decode (*i.e.* recognize) indeterministic phonemes or help in forming a preponderance among probabilistic candidates. In some cases, AVSR may include an audio-

based automatic speech recognition process and an image-based automatic speech recognition process. AVSR may combine results from both processes with feature fusion. Audio-based speech recognition process may analysis audio according to any method described herein, for instance using a Mel frequency cepstral coefficients (MFCCs) and/or log-Mel spectrogram
5 derived from raw audio samples. Image-based speech recognition may perform feature recognition to yield an image vector. In some cases, feature recognition may include any feature recognition process described in this disclosure, for example a variant of a convolutional neural network. In some cases, AVSR employs both an audio datum and an image datum to recognize verbal content. For instance, audio vector and image vector may each be concatenated and used
10 to predict speech made by a user, who is “on camera.”

In some cases, computing device 1804 may be configured to recognize at least a keyword as a function of visual verbal content. In some cases, recognizing at least keyword may include an optical character recognition (OCR). In some cases, computing device 1804 may transcribe much or even substantially all verbal content from target resume video. Similarly for
15 textual resumes, such as written resumes, intelligent character recognition (ICR) may recognize written text one glyph or character at a time, for instance by employing machine-learning processes. In some cases, intelligent word recognition (IWR) may recognize written text, one word at a time, for instance by employing machine-learning processes.

Still referring to FIG. 18, in some cases, OCR may include post-processing. For example,
20 OCR accuracy may be increased, in some cases, if output is constrained by a lexicon. A lexicon may include a list or set of words that are allowed to occur in a document. In some cases, a lexicon may include, for instance, all the words in the English language, or a more technical lexicon for a specific field. In some cases, an output stream may be a plain text stream or file of characters. In some cases, an OCR process may preserve an original layout of visual verbal
25 content. In some cases, near-neighbor analysis can make use of co-occurrence frequencies to correct errors, by noting that certain words are often seen together. For example, “Washington, D.C.” is generally far more common in English than “Washington DOC.” In some cases, an OCR process may make us of a prior knowledge of grammar for a language being recognized. For example, grammar rules may be used to help determine if a word is likely to be a verb or a
30 noun. Distance conceptualization may be employed for recognition and classification. For

example, a Levenshtein distance algorithm may be used in OCR post-processing to further optimize results.

In one or more embodiments, apparatus 1800 may further include a memory component 1840. Memory component 1840 may be communicatively connected to computing device 1804 and may be configured to store information and/or datum related to apparatus 1800, such as posting datum 1812, user datum 1808, information related to record recommendation 1816, information related to interaction preparation 1820, and the like. In one or more embodiments, memory component 1840 is communicatively connected to a processor and configured to contain instructions configuring processor to determine the record recommendation. Memory component 1840 may be configured to store information and datum related to posting match recommendation. For example, memory component 1840 may store previously prepared records (*e.g.*, draft resumes), customized records generated by computing device 1804, posting datum 1812, user datum 1808, interaction preparation 1820, record recommendation 1816, and the like. In one or more embodiments, memory component may include a storage device, as described further in this disclosure below.

With continued reference to FIG. 18, computing device 1804 is configured to provide an interaction preparation 1820. For the purposes of this disclosure, “interaction preparation” is instructions, direction, or interactions used to educate and get ready a user for a job interview related to a job position. A job interview, such as a telephonic or in-person interview, may be conducted by a potential employer that posted the job position listing. Thus, interaction preparation 1820 may simulate an interview process to ready user for an actual interview with the employer of the job position. In one or more embodiments, interaction preparation 1820 may include a simulated interview, which may be done via video, audio, or text. For example, a chatbot may be used to simulate an employer-candidate conversation with the user in real time. A chatbot may include a software application used to conduct a chat conversation using text or speech. Chatbot may include skills that allow chatbot to ask the user various questions related to the job position and user’s qualifications for the job position. Chatbot may generate inquiries as a function of posting datum 1812, user datum 1808, record recommendation 1816, compatibility score 1836, interaction preparation 1820 information, and the like. Chatbot may access database to generate customized inquiries for user and/or to present generic inquiries related to the job

position. Generic inquiries may be imported into database by a user, using a network, using the web, by a third-party application, or other sources of job-related information. In one or more embodiments, display component 1828 component is configured to display interaction preparation 1820. In one or more embodiments, computing device 1804 may implement a
5 chatbot algorithm or a machine-learning module to generate a chatbot.

In one or more embodiments, display component 1828 may be communicatively connected to computing device 1804. Display component may be remote to computing device or integrated into computing device 1804. Communication between computing device 1804 and display component may be wired or wireless. In one or more embodiments, display component
10 1828 may be configured to display posting datum 1812, user datum 1808, record recommendation 1816, interaction preparation 1820, compatibility score 1836, database information, and the like. Display component 1828 may include a graphic user interface (GUI) that a user may use to navigate through presented data or information by computing device 1804. In one or more embodiments, a GUI may include a plurality of lines, images, symbols, and the
15 like to show information and/or data. In non-limiting embodiments, display component 1828 may include a smartphone, tablet, laptop, desktop, monitor, tablet, touchscreen, head-up display (HUD), and the like. In one or more embodiments, display component 1828 may include a screen such as a liquid crystal display (LCD) various other types of displays or monitors, as previously mentioned in this disclosure. In one or more embodiments, user may view
20 information and/or data displayed on display component 1828 in real time. In one or more embodiments, display component may be configured to display received or determined information, which may be toggled through using, for example, an input device of display component or computing device 1804. Display component 1828 may include electronic components utilized to display image data or information, such as a video, GUI, photo, and the
25 like.

Now referring to FIG. 19, a flow chart of method 1900 of digital posting match recommendation is shown. At step 1905, method 1900 includes receiving, using computing device 1804, posting datum 1812. In one or more non-limiting embodiments, posting datum may include information regarding a job position available with a particular employer. These steps
30 may be implemented as described above, without limitation, in FIGS. 1-18.

At step 1910, method 1900 includes determining, by the computing device, a record recommendation 1816 as a function of the posting datum 1812. In one or more embodiments, determining record recommendation 1816 includes generating compatibility score 1836 between user and job position. In one or more embodiments, record recommendation 1816 may include a suggested alteration of a record of a user. In one or more embodiments, method 300 further includes the step of determining record recommendation 1816 by generating a customized record. These steps may be implemented as described above, without limitation, in FIGS. 1-18.

At step 1915, method 1900 includes providing, by computing device, interaction preparation 1820 as a function of record recommendation 1816. In one or more embodiments, the interaction preparation 1820 includes an interactive interview. These steps may be implemented as described above, without limitation, in FIGS. 1-18.

At step 1920, method 300 includes displaying, by display component 1828 record recommendation 1816 or interaction preparation 1820 to user. In one or more embodiments, displaying interaction preparation 1820 includes displaying interaction preparation 1820 using a graphic user interface shown on display component 1828. These steps may be implemented as described above, without limitation, in FIGS. 1-18.

Referring now to FIG. 20, an exemplary embodiment of an apparatus 2000 for linking posting data to a plurality of user identifiers is illustrated. Apparatus 2000 includes a processor 2004 and a memory 2008 communicatively connected to processor 2004, wherein memory 2008 contains instructions configuring processor 2004 to carry out the linking process. Processor 2004 and memory 2008 is contained in a computing device 2012. As used in this disclosure, “communicatively connected” means connected by way of a connection, attachment, or linkage between two or more relata which allows for reception and/or transmittance of information therebetween. For example, and without limitation, this connection may be wired or wireless, direct, or indirect, and between two or more components, circuits, devices, systems, and the like, which allows for reception and/or transmittance of data and/or signal(s) therebetween. Data and/or signals therebetween may include, without limitation, electrical, electromagnetic, magnetic, video, audio, radio, and microwave data and/or signals, combinations thereof, and the like, among others. A communicative connection may be achieved, for example and without limitation, through wired or wireless electronic, digital, or analog, communication, either directly

or by way of one or more intervening devices or components. Further, communicative connection may include electrically coupling or connecting at least an output of one device, component, or circuit to at least an input of another device, component, or circuit. For example, and without limitation, via a bus or other facility for intercommunication between elements of a computing device. Communicative connecting may also include indirect connections via, for example and without limitation, wireless connection, radio communication, low power wide area network, optical communication, magnetic, capacitive, or optical coupling, and the like. In some instances, the terminology “communicatively coupled” may be used in place of communicatively connected in this disclosure. A computing device 2012 may include any computing device as described in this disclosure, including without limitation a microcontroller, microprocessor, digital signal processor (DSP) and/or system on a chip (SoC) as described in this disclosure. Computing device 2012 may include, be included in, and/or communicate with a mobile device such as a mobile telephone or smartphone. Computing device 2012 may include a single computing device operating independently, or may include two or more computing device operating in concert, in parallel, sequentially or the like; two or more computing devices may be included together in a single computing device or in two or more computing devices. Computing device 2012 may interface or communicate with one or more additional devices as described below in further detail via a network interface device. Network interface device may be utilized for connecting computing device 2012 to one or more of a variety of networks, and one or more devices. Examples of a network interface device include, but are not limited to, a network interface card (*e.g.*, a mobile network interface card, a LAN card), a modem, and any combination thereof. Examples of a network include, but are not limited to, a wide area network (*e.g.*, the Internet, an enterprise network), a local area network (*e.g.*, a network associated with an office, a building, a campus or other relatively small geographic space), a telephone network, a data network associated with a telephone/voice provider (*e.g.*, a mobile communications provider data and/or voice network), a direct connection between two computing devices, and any combinations thereof. A network may employ a wired and/or a wireless mode of communication. In general, any network topology may be used. Information (*e.g.*, data, software etc.) may be communicated to and/or from a computer and/or a computing device. Computing device 2012 may include but is not limited to, for example, a computing device or cluster of

computing devices in a first location and a second computing device or cluster of computing devices in a second location. Computing device 2012 may include one or more computing devices dedicated to data storage, security, distribution of traffic for load balancing, and the like. Computing device 2012 may distribute one or more computing tasks as described below across a plurality of computing devices of computing device, which may operate in parallel, in series, redundantly, or in any other manner used for distribution of tasks or memory between computing devices. Computing device 2012 may be implemented using a “shared nothing” architecture in which data is cached at the worker, in an embodiment, this may enable scalability of apparatus 2000 and/or computing device 2012.

With continued reference to FIG. 20, processor 2004 and/or computing device 2012 may be designed and/or configured by memory 2008 to perform any method, method step, or sequence of method steps in any embodiment described in this disclosure, in any order and with any degree of repetition. For instance, processor 2004 and/or computing device 2012 may be configured to perform a single step or sequence repeatedly until a desired or commanded outcome is achieved; repetition of a step or a sequence of steps may be performed iteratively and/or recursively using outputs of previous repetitions as inputs to subsequent repetitions, aggregating inputs and/or outputs of repetitions to produce an aggregate result, reduction or decrement of one or more variables such as global variables, and/or division of a larger processing task into a set of iteratively addressed smaller processing tasks. Processor 2004 and/or computing device 2012 may perform any step or sequence of steps as described in this disclosure in parallel, such as simultaneously and/or substantially simultaneously performing a step two or more times using two or more parallel threads, processor cores, or the like; division of tasks between parallel threads and/or processes may be performed according to any protocol suitable for division of tasks between iterations. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which steps, sequences of steps, processing tasks, and/or data may be subdivided, shared, or otherwise dealt with using iteration, recursion, and/or parallel processing.

Processor 2004 and/or computing device 2012 may perform determinations, classification, and/or analysis steps, methods, processes, or the like as described in this disclosure using machine learning processes 2016. A “machine learning process,” as used in this

disclosure, is a process that automatically uses a body of data known as “training data” and/or a “training set” (described further below) to generate an algorithm that will be performed by a computing device/module to produce outputs given data provided as inputs; this is in contrast to a non-machine learning software program where the commands to be executed are determined in advance by a user and written in a programming language. Machine-learning process 2016 may utilize supervised, unsupervised, lazy-learning processes and/or neural networks, described further below.

Still referring to FIG. 1, processor 2004 and/or computing device 2012 is configured to receive a plurality of user identifiers 2020 relating to a plurality of users, wherein a user identifier 2020 of the plurality of user identifiers 2020 includes at least a video resume. A “user identifier”, as used in this disclosure is information representing a user, including the user’s name, contact information, resume, video resume, Curriculum Vitae, previous work history, cover letter, profile photo, a list of relevant skills, certifications, and the like. “As used in this disclosure, a “user” is a person, for example a jobseeker. User identifier is received by processor 2004 and/or computing device 2012 in at least the form of a video resume. A “video resume” is an item of digital media in visual and/or audio form to provide a recording promoting a user. In some cases, the video resume may include content that is representative or communicative of at least an attribute of the user. Attributes may include information representing a user such as skills, competencies, experience, credentials, talents, and the like. The user may be represented directly by the video resume. For example, in some cases, an image component may represent a visual of the user. As used in this disclosure, an “image component” may be a visual representation of information, such as a plurality of temporally sequential frames and/or pictures, related to the video resume. For example, an image component may include animations, still imagery, recorded video, and the like. In some embodiments, the video resume may be an uploaded to a user database 2024 by a user computing device operated by a user that is communicatively connected to processor 2004 and/or computing device 2012 through a network. As non-limiting examples, the user computing device may be a smartphone, tablet, laptop, pc, and the like.

With continued reference to FIG. 20, a “user database” is a resource storage system used to collect and store any information received from a user, such as videos, images, documents,

and the like. User database 2024 may be implemented, without limitation, as a relational database, a key-value retrieval database such as a NOSQL database, or any other format or structure for use as a database that a person skilled in the art would recognize as suitable upon review of the entirety of this disclosure. User database 2024 may alternatively or additionally be
5 implemented using a distributed data storage protocol and/or data structure, such as a distributed hash table or the like. User database 2024 may include a plurality of data entries and/or records as described above. Data entries in a database may be flagged with or linked to one or more additional elements of information, which may be reflected in data entry cells and/or in linked tables such as tables related by one or more indices in a relational database. Persons skilled in the
10 art, upon reviewing the entirety of this disclosure, will be aware of various ways in which data entries in a database may store, retrieve, organize, and/or reflect data and/or records as used herein, as well as categories and/or populations of data consistently with this disclosure.

With continued reference to FIG. 20, in some cases, the video resume may include a digital video, which may be communicated by way of digital signals, for example between
15 computing devices which are communicatively connected with at least a network. To optimize speed and/or cost of transmission of the video resume. Video may be compressed according to a video compression coding format (i.e., codec). Exemplary video compression codecs include H.26x codecs, MPEG formats, SVT-AV1, and the like. In some cases, compression of a digital video may be lossy, in which some information may be lost during compression. Alternatively,
20 or additionally, in some cases, compression of a digital video may be substantially lossless, where substantially no information is lost during compression.

Still referring to FIG. 20, the video resume may be representative of subject-specific data. As used in this disclosure, “subject-specific data” is any element of information that is associated with a specific user. Exemplary forms of subject-specific data include an image component, the
25 video resume, non-verbal content, verbal content, audio component, as well as any information derived directly or indirectly from the video resume or any other subject-specific data. For example, subject-specific data could be the physical properties of a user, such as their body posture or facial expression. Subject-specific data could also be audio sensory properties of a user, such as tone of voice or background audio in the video resume. More in-depth examples
30 and descriptions of subject-specific data are illustrated in U.S. Patent Application No.

207/582,070, filed on January 24, 2022, and entitled “SYSTEMS AND METHODS FOR PARSING AND COMPARING VIDEO RECORD DUPLICATIONS”, the entirety of which is incorporated by reference in this disclosure.

With continued reference to FIG. 20, in some embodiments, an image component may include or otherwise represent verbal content. For instance, written or visual verbal content may be included within an image component. Visual verbal content may include images of written text represented by an image component. For example, visual verbal content may include, without limitation, digitally generated graphics, images of written text (e.g., typewritten, and the like), signage, and the like.

Still referring to FIG. 20, in some embodiments, an image component may include or otherwise represent audible verbal content related to at least an attribute of a user. As used in this disclosure, “audible verbal content” is oral (e.g., spoken) verbal content. In some cases, audible verbal content may be included within the video resume by way of an audio component. As used in this disclosure, an “audio component” is a representation of audio, for example a sound, a speech, and the like. In some cases, verbal content may be related to at least an attribute of subject. Additionally, or alternatively, visual verbal content and audible verbal content may be used as inputs to classifier 2044 as described throughout this disclosure.

In some cases, processor 2004 and/or computing device 2012 may include using audiovisual speech recognition (AVSR) processes to recognize verbal content in the video resume. For example, processor 2004 and/or computing device 2012 may use image content to aid in recognition of audible verbal content such as viewing a user move their lips to speak on video to process the audio content of the video resume. AVSR may use an image component to aid the overall translation of the audio verbal content of the video resume. In some embodiments, AVSR may include techniques employing image processing capabilities in lip reading to aid speech recognition processes. In some cases, AVSR may be used to decode (i.e., recognize) indeterministic phonemes or help in forming a preponderance among probabilistic candidates. In some cases, AVSR may include an audio-based automatic speech recognition process and an image-based automatic speech recognition process. AVSR may combine results from both processes with feature fusion. Audio-based speech recognition process may analysis audio according to any method described herein, for instance using a Mel frequency cepstrum

coefficients (MFCCs) and/or log-Mel spectrogram derived from raw audio samples. Image-based speech recognition may perform feature recognition to yield an image vector. In some cases, feature recognition may include any feature recognition process described in this disclosure, for example a variant of a convolutional neural network. In some cases, AVSR employs both an audio datum and an image datum to recognize verbal content. For instance, audio vector and image vector may each be concatenated and used to predict speech made by a user, who is ‘on camera.’ Other applicable methods of acquiring verbal content are illustrated in U.S. Patent Application No. 17/582,070, filed on January 24, 2022, and entitled “SYSTEMS AND METHODS FOR PARSING AND COMPARING VIDEO RECORD DUPLICATIONS”, the entirety of which is incorporated by reference in this disclosure.

Still referring to FIG. 20, processor 2004 and/or computing device 2012 is configured to receive posting data 2028 from a posting generator of a plurality of posting generators. As used in this disclosure, “posting data” is information pertaining to the hiring requirements, job descriptions, preferences, expected salary, relocation considerations, employment timeline, and the like of a posting generator. A “posting generator” is a hiring entity or employer. For example, the posting generator could be a company, job matching service, and the like. In some embodiments, posting data 2028 may be stored in a posting database 2032 connected to computing device 2012 using any network interface described throughout this disclosure. As used in this disclosure, a “posting database” is a database containing a plurality of posting data 2028 from numerous or the same posting generator. Posting database 2032 may be implemented, without limitation, as a relational database, a key-value retrieval database such as a NOSQL database, or any other format or structure for use as a database that a person skilled in the art would recognize as suitable upon review of the entirety of this disclosure. Posting database 2032 may alternatively or additionally be implemented using a distributed data storage protocol and/or data structure, such as a distributed hash table or the like. Posting database 2032 may include a plurality of data entries and/or records as described above. Data entries in a database may be flagged with or linked to one or more additional elements of information, which may be reflected in data entry cells and/or in linked tables such as tables related by one or more indices in a relational database. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which data entries in a database may store, retrieve, organize, and/or

reflect data and/or records as used herein, as well as categories and/or populations of data consistently with this disclosure.

Still referring to FIG. 20, processor 2004 and/or computing device 2012 is configured to identify a plurality of keywords 2040 within posting data 2028. Posting data 2028 may be in the form of a document, pdf, HTML, HTM, text file (TXT) and the like. As used in this disclosure, a “keyword” is an element of word or syntax used to identify and/or match elements to each other. Keywords 2040 may refer to the descriptive elements of posting data 2028 related to a posting generator. In some embodiments, a plurality of keywords 2040 may be based on a posting category. A “posting category,” as used in this disclosure, is a specific grouping of related information. For example, posting data 2028 may be a job listing wherein information is sectioned off under specific headings, such as job description, hiring requirements, geographic location, and the like. In some cases, processor 2004 and/or computing device 2012 may be configured to recognize at least a keyword in a video resume as a function of visual verbal content. In some cases, recognizing a plurality of keywords 2040 in posting data 2028 may include a language processing module 2036. In some embodiments, keywords 2040 may be extracted as the descriptive elements of a plurality of posting categories contained in posting data 2028. In some embodiments, samples, and examples of keywords 2040 submitted by a hiring entity or apparatus 2000 administrator may be use used to train language processing module 2036 in identifying keywords 2028. For instance, a sample may be a list of synonyms used for common words used by hiring entities and job posting, such as “advocate,” “attorney-at-law,” “counsel,” “counselor”, “counselor-at-law”, “lawyer”, and “legal eagle”. These samples and examples may additionally be used to classify keywords 2040 to similar keywords contained in the plurality of user identifiers 2024, as described further below. Language processing module 2036 may include any hardware and/or software module. For example, the module may be used to extract from posting data 2028 all information pertaining to “truck driver” of posting data 2028. Language processing module 2036 may be implemented in any manner described in this disclosure.

Still referring to FIG. 20, in some cases, recognizing keywords in posting data 2028 and video resume may include using optical character recognition, which may be performed without limitation in any manner described in this disclosure.

Still referring to FIG. 20, processor 2004 and/or computing device 2012 is configured generate a keyword ranking of posting data 2028 based on a plurality of weighted values reflecting priority among the plurality of keywords 2040. As used in this disclosure, a “keyword ranking” is the plurality of keywords extracted from posting data ordered in descending level of priority. As used in the disclosure, “priority” is the state of importance of a keyword in order of ranking. Priority may be based on the job fulfillment needs of the posting generator. For example, keywords 2040 related to education requirements may be given a higher weighted value than keywords 2040 related to hobbies. Generating ranking may include using a machine learning process 2016 including a feedback loop to generate the plurality of weighted values based on training data containing at least priority feedback from the plurality of posting generators. As used in this disclosure, a “feedback loop” is the process by which a machine learning module/module/algorithm output is routed back as inputs as part of a chain of cause-and-effect that forms a circuit or loop. This may allow the machine learning system to adjust weighted values for optimized ranking output in the future. “Priority feedback” is samples of positive or negative keyword rankings based on priority. In some embodiments, priority feedback may be contained in posting database 2032 or a priority feedback database that a plurality of posting generators can upload positive or negative samples to. In some embodiments, the priority feedback from the plurality of posting generators may be received from the priority feedback database. Priority feedback database may be implemented, without limitation, as a relational database, a key-value retrieval database such as a NOSQL database, or any other format or structure for use as a database that a person skilled in the art would recognize as suitable upon review of the entirety of this disclosure. Priority feedback database may alternatively or additionally be implemented using a distributed data storage protocol and/or data structure, such as a distributed hash table or the like. Priority feedback database may include a plurality of data entries and/or records as described above. Data entries in a database may be flagged with or linked to one or more additional elements of information, which may be reflected in data entry cells and/or in linked tables such as tables related by one or more indices in a relational database. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which data entries in a database may store, retrieve, organize, and/or

reflect data and/or records as used herein, as well as categories and/or populations of data consistently with this disclosure.

In some embodiments, as a non-limiting example, processor 2004 and/or computing device 2012 may be configured to display keyword ranking to a posting generator operated
5 display device 2052, the posting generator may then leave feedback on the ranking that is to be submitted to the priority feedback database. A "display device", as used herein is an output device for presentation of information in visual or tactile form. In one or more embodiments, display device 2048 may be communicatively connected to computing device 2012. Display device 2048 may be remote to computing device or integrated into computing device 2012.
10 Communication between computing device 2012 and display device 2048 may be wired or wireless. As used in this disclosure, "feedback" is the transmission of evaluative or corrective information about an action, event, or process to the original or controlling source. In some embodiments, display device 2048 may include an LCD display, an OLED display, a CRT display, an LED display, and the like. The posting generator feedback may be as simple as
15 marking the ranking as good or bad or may be detailed guidelines to how each keyword of the plurality of keywords 2040 should be weighted. For example, processor 2004 and/or computing device 2012 may be configured to generate a form for a posting generator to fill out, asking questions such as, "Was this ranking to your satisfaction, if not explain what should be different", "List examples of keywords to be included", and the like. The form may be outputted
20 as an attachment to the keyword ranking. Additionally, feedback may be in the form of manually adjusting the ranking of the keywords. In some embodiments, the posting generator may provide a ranking score. A "ranking score" is a score indicating a level of satisfaction or dissatisfaction with the ranking. As non-limiting examples, the ranking score may be given as a letter grade (i.e. A to F), a percentage, a decimal, a fraction, or a score out of ten. As part of the training data, a
25 machine learning algorithm, which may be a component of machine learning processes 2016 may then take the original ranked keyword output as a new input and generate a better keyword ranking based on the priority feedback. This feedback process may used to further train the machine learning system to learn priority patterns among a plurality of posting generators when determining weighted values. As a non-limiting example, the machine learning system may learn

that, as a general rule, posting generators prioritize keywords 2040 related to work experience over keywords 2040 related to education and should be weighted accordingly.

Still referring to FIG. 20, processor 2004 and/or computing device 2012 is configured to match a plurality of keywords 2040 of the keyword ranking to the plurality of user identifiers

5 2020. In some embodiments, matching may include classifying the keywords 2040 contained in posting data 2028 to similar keywords contained in a video resume. For example, keywords 2040 relating “web developer internship experience” may be matched to the plurality of video resumes containing similar keywords, similar keywords may be based on synonyms of keywords 2040 as described above. Matching may occur through a classifier 2044. Classifier 2044 may be

10 configured to output at least a datum that labels or otherwise identifies a set of data that are clustered together, found to be close under a distance metric as described below, or the like.

Processor 2004 and/or computing device 2012 and/or another device may generate classifier

2044 using a classification algorithm, defined as a processes whereby a processor 2004 and/or computing device 2012 derives classifier 2044 from training data. In an embodiment, training

15 data may include data from User database 2024, sample and examples of keywords 2040, posting database 2032, language processing module 2036, and any other training data described

throughout this disclosure. Classifier 2044 may take the plurality of ranked keywords 2040 from posting data 2028 and the plurality of user identifiers 2020 contained in user database 2024 as

algorithm inputs. Classifier 2044 may then use the training data disclosed above to output data

20 bins of particular user identifiers 2020 of the plurality of user identifiers 2020 matched to posting data 2028. Each data bin may be categorized to each user identifier 2020 contained in User

database 2024 and labeled with the contact information of the user. Classification may be performed using, without limitation, linear classifiers such as without limitation logistic

25 regression and/or naive Bayes classifiers, nearest neighbor classifiers such as k-nearest neighbors classifiers, support vector machines, least squares support vector machines, fisher’s linear

discriminant, quadratic classifiers, decision trees, boosted trees, random forest classifiers, learning vector quantization, and/or neural network-based classifier 2044s.

Still referring to FIG. 20, processor 2004 and/or computing device 2012 is configured to generate, as a function of the matching, a ranking of the plurality of user identifiers 2020 using a

30 fuzzy set inference system 2052 based on a superiority criterion of each user identifier. In some,

embodiments, ranking may be based on the superiority criterion of each user identifier, regarding compatibility to the posting data 2028, compared to the plurality of user identifiers 2020. For example, “John Smith” may be ranked higher than “Jane Doe” based on a determination that Smith’s resume is superior to Doe’s based on the high compatibility of Smith’s resume to a job posting. In some embodiments, compatibility may be based on the similarity of keywords in user identifier 2020 to keywords 2040 contained in posting data 2028. In some embodiments, compatibility may be based on the total amount of similar keywords contained in user identifier 2020. In some embodiments, compatibility may be the quality of keywords found in user identifier 2020, for example, Smith’s resume may contain keywords that are prioritized higher than the keywords contained in Doe’s. Ranking the plurality of user identifiers 2020 includes using a fuzzy set inference system 2052 as discussed further below. Processor 2004 and/or computing device 2012 may take the output data of classifier 2044 and use it as fuzzy set input wherein the inference system may utilize a superiority criterion to rank each matched user identifier against that the plurality of matched user identifiers 2020. In some embodiments, as a function of the user identifier ranking, processor 2004 and/or computing device 2012 may be configured to send the inference system output to display device 2048 operated by the posting generator associated with the posting data 2028.

Referring now to FIG. 21, in an overview of an exemplary method 2100 for linking posting data to a plurality of user identifiers. Method 2100 includes using a computing device, being any described through this disclosure, for example and with reference to FIGS. 1-20. At step 2105, method 2100 includes using a computing device configured to receive a plurality of user identifiers relating to a plurality of users, wherein a user identifier of the plurality of user identifiers includes at least a video resume, for example and with reference to FIG. 20. The plurality of user identifiers is received by the computing in a at least a video resume format. The computing device may receive such data through a user database as described in FIG. 20. In some cases, the video resume may include content that is representative or communicative of at least an attribute of the user. Attributes may include information representing a user such as skills, competencies, experience, credentials, talents, and the like. The user may be represented directly by video resume. The video resume may contain content, components, and data, as described in FIG. 20, such as verbal content.

In some cases, the video resume may include a digital video, which may be communicated by way of digital signals, for example between computing devices which are communicatively connected with at least a network. To optimize speed and/or cost of transmission of the video resume. Video may be compressed according to a video compression coding format (i.e., codec). Exemplary video compression codecs include H.26x codecs, MPEG formats, SVT-AV1, and the like. In some cases, compression of a digital video may be lossy, in which some information may be lost during compression. Alternatively, or additionally, in some cases, compression of a digital video may be substantially lossless, where substantially no information is lost during compression.

Still referring to FIG. 21, in some embodiments, an image component may include or otherwise represent audible verbal content related to at least an attribute of a user. As used in this disclosure, “audible verbal content” is oral (e.g., spoken) verbal content. In some cases, audible verbal content may be included within the video resume by way of an audio component. As used in this disclosure, an “audio component” is a representation of audio, for example a sound, a speech, and the like. In some cases, verbal content may be related to at least an attribute of subject. Additionally, or alternatively, visual verbal content and audible verbal content may be used as inputs to classifiers as described throughout this disclosure.

In some cases, the computing device may include audiovisual speech recognition (AVSR) processes to recognize verbal content in the video resume. For example, and with reference to FIG. 20, the computing device may use image content to aid in recognition of audible verbal content such as viewing a user move their lips to speak on video to process the audio content of the video resume. AVSR may use an image component to aid the overall translation of the audio verbal content of the video resume. In some embodiments, AVSR may include techniques employing image processing capabilities in lip reading to aid speech recognition processes.

Still referring to FIG. 21, at step 2110, method 2100 includes using a computing device configured to receive posting data from a posting generator of a plurality of posting generators, for example and with reference to FIG. 20. Posting data may include a plurality of descriptive data elements pertaining to the posting generator. In some embodiments, the posting data may be stored in a database connected to the computing device using any network interface described

throughout this disclosure. In some embodiments, the posting data may be stored in a database connected to the computing device using any network interface described throughout this disclosure. The user database and the posting database may be implemented, without limitation, as a relational database, a key-value retrieval database such as a NOSQL database, or any other
5 format or structure for use as a database that a person skilled in the art would recognize as suitable upon review of the entirety of this disclosure. The database may alternatively or additionally be implemented using a distributed data storage protocol and/or data structure, such as a distributed hash table or the like. The database may include a plurality of data entries and/or records as described above. Data entries in a database may be flagged with or linked to one or
10 more additional elements of information, which may be reflected in data entry cells and/or in linked tables such as tables related by one or more indices in a relational database. Persons skilled in the art, upon reviewing the entirety of this disclosure, will be aware of various ways in which data entries in a database may store, retrieve, organize, and/or reflect data and/or records as used herein, as well as categories and/or populations of data consistently with this disclosure.

15 Still referring to FIG. 21, at step 2115, method 2100 includes using a computing device configured to identify a plurality of keywords within the posting data, for example and with reference to FIG 20. The plurality of keywords may include at least the plurality of descriptive data elements of the posting generator. Varying embodiments and example of the plurality of keywords are reference here as described in FIG. 20. The posting data may be in the form of a
20 document, pdf, HTML, HTM, text file (TXT) and the like. In some cases, the computing device may be configured to recognize at least a keyword in a video resume as a function of visual verbal content. In some cases, recognizing keywords in the posting data may include a language processing module, for example and with reference to FIG. 20. A language processing module may include any hardware and/or software module. In some cases, recognizing keywords in the
25 posting data and video resume may include using optical character recognition, for example and with reference to FIG.20.

At step 2120, method 2100 includes using a computing device configured to generate a keyword ranking of posting data based on a plurality of weighted values reflecting priority among the plurality of keywords, for example and with reference to FIG. 20. Priority may be
30 based on the job fulfillment needs of the posting generator. For example, keywords related to

education requirements many given a higher weighted value than keywords related to hobbies. Generating ranking may include using a machine learning process, as described in FIGS. 1-20, including a feedback loop to generate the plurality of weighted values based on training data containing at least priority feedback from the plurality of posting generators. This may allow the machine learning system to adjust weighted values for optimized ranking output in the future. In some embodiments, priority feedback may be contained in a priority feedback database, as any described throughout this disclosure, that a plurality of posting generators can upload positive or negative samples to. In some embodiments, the at least priority feedback from the plurality of posting generators is received from a priority feedback database containing a plurality of samples related to the keyword ranking. This feedback process may be used by training the machine learning system to learn keyword priority patterns among a plurality of posting generators when determining weighted values.

At step 2125, method 2100 includes using a computing device configured to match a plurality of keywords of the keyword ranking to the plurality of user identifiers, for example and with reference to FIG. 20. In some embodiments, matching may include classifying the keywords containing in posting data to similar keywords contained in video resume. Matching may occur through a classifier as described in FIGS. 1-20. In an embodiment, classifier training data may include the user database, the posting data database, samples, and examples of keywords a language processing module, and any other training data described throughout this disclosure. The classifier may take the plurality of ranked keywords from posting data and the plurality of user identifiers contained in the user database as algorithm inputs. The classifier may then use the training data disclosed above to output data bins of particular user identifiers of the plurality of user identifiers matched to the posting data, for example and with reference to FIG. 20. Each data bin may be labeled with the contact information of the user. Classification may be performed using, without limitation, linear classifiers such as without limitation logistic regression and/or naive Bayes classifiers, nearest neighbor classifiers such as k-nearest neighbors classifiers, support vector machines, least squares support vector machines, fisher's linear discriminant, quadratic classifiers, decision trees, boosted trees, random forest classifiers, learning vector quantization, and/or neural network-based classifiers.

Still referring to FIG. 21, at step 2130, method 2100 includes using a computing device configured to generate, as a function of the matching, a ranking of the plurality of user identifiers based on a superiority criterion using a fuzzy set inference system. In some embodiments, ranking may be based on the superiority of each user identifier, regarding compatibility to the posting data, compared to the plurality of user identifiers. Ranking of the plurality of user identifiers includes using a fuzzy set inference system as discussed in this disclosure. The computing device may take the output data of classifier and use it as fuzzy set input where in the inference system may utilize a superiority criterion to rank each matched user identifier against that the plurality of matched user identifiers. In some embodiments, as a function of the user identifier ranking, the computing device may be configured to send the inference system output to a computing device operated by the posting generator associated with the posting data.

Examples of a computing device include, but are not limited to, an electronic book reading device, a computer workstation, a terminal computer, a server computer, a handheld device (e.g., a tablet computer, a smartphone, etc.), a web appliance, a network router, a network switch, a network bridge, any machine capable of executing a sequence of instructions that specify an action to be taken by that machine, and any combinations thereof. In one example, a computing device may include and/or be included in a kiosk.

It is to be noted that any one or more of the aspects and embodiments described herein may be conveniently implemented using digital electronic circuitry, integrated circuitry, specially designed application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs) computer hardware, firmware, software, and/or combinations thereof, as realized and/or implemented in one or more machines (e.g., one or more computing devices that are utilized as a user computing device for an electronic document, one or more server devices, such as a document server, etc.) programmed according to the teachings of the present specification, as will be apparent to those of ordinary skill in the computer art. These various aspects or features may include implementation in one or more computer programs and/or software that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device. Appropriate software coding may readily be prepared by skilled

programmers based on the teachings of the present disclosure, as will be apparent to those of ordinary skill in the software art. Aspects and implementations discussed above employing software and/or software modules may also include appropriate hardware for assisting in the implementation of the machine executable instructions of the software and/or software module.

5 Such software may be a computer program product that employs a machine-readable storage medium. A machine-readable storage medium may be any medium that is capable of storing and/or encoding a sequence of instructions for execution by a machine (e.g., a computing device) and that causes the machine to perform any one of the methodologies and/or
10 embodiments described herein. Examples of a machine-readable storage medium include, but are not limited to, a magnetic disk, an optical disc (e.g., CD, CD-R, DVD, DVD-R, etc.), a magneto-optical disk, a read-only memory “ROM” device, a random access memory “RAM” device, a magnetic card, an optical card, a solid-state memory device, an EPROM, an EEPROM, Programmable Logic Devices (PLDs), and/or any combinations thereof. A machine-readable
15 medium, as used herein, is intended to include a single medium as well as a collection of physically separate media, such as, for example, a collection of compact discs or one or more hard disk drives in combination with a computer memory. As used herein, a machine-readable storage medium does not include transitory forms of signal transmission.

 Such software may also include information (e.g., data) carried as a data signal on a data carrier, such as a carrier wave. For example, machine-executable information may be included as
20 a data-carrying signal embodied in a data carrier in which the signal encodes a sequence of instruction, or portion thereof, for execution by a machine (e.g., a computing device) and any related information (e.g., data structures and data) that causes the machine to perform any one of the methodologies and/or embodiments described herein.

 Examples of a computing device include, but are not limited to, an electronic book
25 reading device, a computer workstation, a terminal computer, a server computer, a handheld device (e.g., a tablet computer, a smartphone, etc.), a web appliance, a network router, a network switch, a network bridge, any machine capable of executing a sequence of instructions that specify an action to be taken by that machine, and any combinations thereof. In one example, a computing device may include and/or be included in a kiosk.

FIG. 22 shows a diagrammatic representation of one embodiment of a computing device in the exemplary form of a computer system 2200 within which a set of instructions for causing a control system to perform any one or more of the aspects and/or methodologies of the present disclosure may be executed. It is also contemplated that multiple computing devices may be utilized to implement a specially configured set of instructions for causing one or more of the devices to perform any one or more of the aspects and/or methodologies of the present disclosure. Computer system 2200 includes a processor 2204 and a memory 2208 that communicate with each other, and with other components, via a bus 2212. Bus 2212 may include any of several types of bus structures including, but not limited to, a memory bus, a memory controller, a peripheral bus, a local bus, and any combinations thereof, using any of a variety of bus architectures.

Memory 2208 may include various components (*e.g.*, machine-readable media) including, but not limited to, a random-access memory component, a read only component, and any combinations thereof. In one example, a basic input/output system 2216 (BIOS), including basic routines that help to transfer information between elements within computer system 2200, such as during start-up, may be stored in memory 2208. Memory 2208 may also include (*e.g.*, stored on one or more machine-readable media) instructions (*e.g.*, software) 2220 embodying any one or more of the aspects and/or methodologies of the present disclosure. In another example, memory 2208 may further include any number of program modules including, but not limited to, an operating system, one or more application programs, other program modules, program data, and any combinations thereof.

Computer system 2200 may also include a storage device 2224. Examples of a storage device (*e.g.*, storage device 2224) include, but are not limited to, a hard disk drive, a magnetic disk drive, an optical disc drive in combination with an optical medium, a solid-state memory device, and any combinations thereof. Storage device 2224 may be connected to bus 2212 by an appropriate interface (not shown). Example interfaces include, but are not limited to, SCSI, advanced technology attachment (ATA), serial ATA, universal serial bus (USB), IEEE 1394 (FIREWIRE), and any combinations thereof. In one example, storage device 2224 (or one or more components thereof) may be removably interfaced with computer system 2200 (*e.g.*, via an external port connector (not shown)). Particularly, storage device 2224 and an associated

machine-readable medium 2228 may provide nonvolatile and/or volatile storage of machine-readable instructions, data structures, program modules, and/or other data for computer system 2200. In one example, software 2220 may reside, completely or partially, within machine-readable medium 2228. In another example, software 2220 may reside, completely or partially,
5 within processor 2204.

Computer system 2200 may also include an input device 2232. In one example, a user of computer system 2200 may enter commands and/or other information into computer system 2200 via input device 2232. Examples of an input device 2232 include, but are not limited to, an alpha-numeric input device (*e.g.*, a keyboard), a pointing device, a joystick, a gamepad, an audio
10 input device (*e.g.*, a microphone, a voice response system, etc.), a cursor control device (*e.g.*, a mouse), a touchpad, an optical scanner, a video capture device (*e.g.*, a still camera, a video camera), a touchscreen, and any combinations thereof. Input device 2232 may be interfaced to bus 2212 via any of a variety of interfaces (not shown) including, but not limited to, a serial interface, a parallel interface, a game port, a USB interface, a FIREWIRE interface, a direct
15 interface to bus 2212, and any combinations thereof. Input device 2232 may include a touch screen interface that may be a part of or separate from display 2236, discussed further below. Input device 2232 may be utilized as a user selection device for selecting one or more graphical representations in a graphical interface as described above.

A user may also input commands and/or other information to computer system 2200 via
20 storage device 2224 (*e.g.*, a removable disk drive, a flash drive, etc.) and/or network interface device 2240. A network interface device, such as network interface device 2240, may be utilized for connecting computer system 2200 to one or more of a variety of networks, such as network 2244, and one or more remote devices 2248 connected thereto. Examples of a network interface device include, but are not limited to, a network interface card (*e.g.*, a mobile network interface
25 card, a LAN card), a modem, and any combination thereof. Examples of a network include, but are not limited to, a wide area network (*e.g.*, the Internet, an enterprise network), a local area network (*e.g.*, a network associated with an office, a building, a campus or other relatively small geographic space), a telephone network, a data network associated with a telephone/voice provider (*e.g.*, a mobile communications provider data and/or voice network), a direct connection
30 between two computing devices, and any combinations thereof. A network, such as network

2244, may employ a wired and/or a wireless mode of communication. In general, any network topology may be used. Information (e.g., data, software 2220, etc.) may be communicated to and/or from computer system 2200 via network interface device 2240.

Computer system 2200 may further include a video display adapter 2252 for
5 communicating a displayable image to a display device, such as display device 2236. Examples of a display device include, but are not limited to, a liquid crystal display (LCD), a cathode ray tube (CRT), a plasma display, a light emitting diode (LED) display, and any combinations thereof. Display adapter 2252 and display device 2236 may be utilized in combination with processor 2204 to provide graphical representations of aspects of the present disclosure. In
10 addition to a display device, computer system 2200 may include one or more other peripheral output devices including, but not limited to, an audio speaker, a printer, and any combinations thereof. Such peripheral output devices may be connected to bus 2212 via a peripheral interface 2256. Examples of a peripheral interface include, but are not limited to, a serial port, a USB connection, a FIREWIRE connection, a parallel connection, and any combinations thereof.

15 The foregoing has been a detailed description of illustrative embodiments of the invention. Various modifications and additions can be made without departing from the spirit and scope of this invention. Features of each of the various embodiments described above may be combined with features of other described embodiments as appropriate in order to provide a multiplicity of feature combinations in associated new embodiments. Furthermore, while the
20 foregoing describes a number of separate embodiments, what has been described herein is merely illustrative of the application of the principles of the present invention. Additionally, although particular methods herein may be illustrated and/or described as being performed in a specific order, the ordering is highly variable within ordinary skill to achieve embodiments as disclosed herein. Accordingly, this description is meant to be taken only by way of example, and
25 not to otherwise limit the scope of this invention.

In the descriptions above and in the claims, phrases such as “at least one of” or “one or more of” may occur followed by a conjunctive list of elements or features. The term “and/or” may also occur in a list of two or more elements or features. Unless otherwise implicitly or explicitly contradicted by the context in which it is used, such a phrase is intended to mean any
30 of the listed elements or features individually or any of the recited elements or features in

combination with any of the other recited elements or features. For example, the phrases “at least one of A and B;” “one or more of A and B;” and “A and/or B” are each intended to mean “A alone, B alone, or A and B together.” A similar interpretation is also intended for lists including three or more items. For example, the phrases “at least one of A, B, and C;” “one or more of A,
5 B, and C;” and “A, B, and/or C” are each intended to mean “A alone, B alone, C alone, A and B together, A and C together, B and C together, or A and B and C together.” In addition, use of the term “based on,” above and in the claims is intended to mean, “based at least in part on,” such that an unrecited feature or element is also permissible.

The subject matter described herein can be embodied in systems, apparatus, methods, and/or
10 articles depending on the desired configuration. The implementations set forth in the foregoing description do not represent all implementations consistent with the subject matter described herein. Instead, they are merely some examples consistent with aspects related to the described subject matter. Although a few variations have been described in detail above, other modifications or additions are possible. In particular, further features and/or variations can be
15 provided in addition to those set forth herein. For example, the implementations described above can be directed to various combinations and sub-combinations of the disclosed features and/or combinations and sub-combinations of several further features disclosed above. In addition, the logic flows depicted in the accompanying figures and/or described herein do not necessarily require the particular order shown, or sequential order, to achieve desirable results. Other
20 implementations may be within the scope of the following claims.

WHAT IS CLAIMED IS:

1. An apparatus for automated video record generation, the apparatus comprising:
a plurality of sensors configured to:
capture at least audiovisual data from a user;
5 generate at least an audiovisual datum based on the at least audiovisual data from
a user;
at least a processor communicatively connected to the plurality of sensors; and
a memory communicatively connected to the processor, the memory containing
instructions configuring the at least a processor to:
10 convert the at least an audiovisual datum into a video file; and
generate a video record of a user based on the at least an audiovisual datum,
wherein generating the video record further comprises:
identifying a plurality of temporal sections of the video file;
classifying each temporal section of the plurality of temporal sections to a
15 record prompt of a plurality of record prompts, where the plurality
of record prompts is arranged in a prompt ordering; and
assembling the plurality of classified temporal sections into the video
record using the prompt ordering.
2. The apparatus of claim 1, wherein the plurality of sensors comprises an auditory sensor.
- 20 3. The apparatus of claim 1, wherein the plurality of sensors comprises an optical sensor.
4. The apparatus of claim 1, wherein the processor is further configured to transcribe the
video record.
5. The apparatus of claim 1, wherein the processor is further configured to train a neural
network to identify a plurality of temporal sections of the video file.
- 25 6. The apparatus of claim 1, wherein the processor is further configured to use a machine-
learning module to generate record prompts.
7. The apparatus of claim 6, wherein the machine-learning module is further configured to
use a classifier to classify job posting data generate record prompts.
8. The apparatus of claim 1, wherein the processor further comprises a display component.
- 30 9. The apparatus of claim 8, wherein the display component is configured to display the

video record.

10. The apparatus of claim 1, wherein the video record comprises a written component.

11. A method for automated video record generation, the method comprising:

capturing, by plurality of sensors, at least audiovisual data from a user;

5 generating, by plurality of sensors, at least an audiovisual datum;

converting, by processor, the at least an audiovisual datum into a video file;

generating, by processor, a video record of a user based on the at least an audiovisual datum, wherein assembling the video record further comprises:

identifying, by processor, a plurality of temporal sections of the video file;

10 classifying, by processor, each temporal section of the plurality of temporal sections to a record prompt of a plurality of record prompts, where the plurality of record prompts is arranged in a prompt ordering; and assembling, by processor, the plurality of classified temporal sections into the video record using the prompt ordering.

15 12. The method of claim 11, wherein the plurality of sensors comprises an auditory sensor.

13. The method of claim 11, wherein the plurality of sensors comprises an optical sensor.

14. The method of claim 11, wherein the processor is further configured to transcribe the video record.

20 15. The method of claim 11, wherein the processor is further configured to train a neural network to identify a plurality of temporal sections of the video file.

16. The method of claim 11, wherein the processor is further configured to use a machine-learning module to generate record prompts.

17. The method of claim 16, wherein the machine-learning module is further configured to use a classifier to classify job posting data generate record prompts.

25 18. The method of claim 11, wherein the processor further comprises a display component.

19. The method of claim 18, wherein the display component is configured to display the video record.

20. The method of claim 11, wherein the video record comprises a written component.

21. An apparatus for creating a video record, the apparatus comprising:

30 at least a processor communicatively connected to a user device; and

a memory communicatively connected to the processor, the memory containing instructions configuring the at least a processor to:
prompt a user with an initial request, wherein prompting the user with an initial request comprises determining the initial request based on a posting;

5 receive an initial response from the user;

prompt the user with a subsequent request based on the initial response;

receive a subsequent response from the user; and

create a video record as a function of the initial response and the subsequent response.

22. The apparatus of claim 21, wherein the processor is configured to determine the initial
10 request based on a user document.

23. The apparatus of claim 21, wherein the processor is configured to determine the initial request based on a keyword selected by the user.

24. The apparatus of claim 21, wherein the processor is configured to determine the subsequent request based on the posting.

15 25. The apparatus of claim 21, wherein the initial response and the subsequent response comprise a visual component.

26. The apparatus of claim 21, wherein the processor, utilizing a machine-learning module, is configured to output the subsequent request based on an input of the initial response.

27. The apparatus of claim 26, wherein the processor is configured to determine a category
20 corresponding to the user.

28. The apparatus of claim 27, wherein the subsequent request is associated with the category.

29. The apparatus of claim 21, wherein the video record comprises an audio recording of the initial response and the subsequent response.

25 30. The apparatus of claim 21, wherein the processor is configured to determine a score based on a duration of the initial response and a duration of the subsequent response.

31. A method for creating a video record, the method comprising:

prompting, by a processor, a user with an initial request;

receiving, at the processor, an initial response from the user;

30 prompting, by the processor, the user with a subsequent request based on the initial

response, wherein prompting the user with an initial request comprises
determining the initial request based on a posting;
receiving, at the processor, a subsequent response from the user; and
creating, by the processor, a video record as a function of the initial response and the
5 subsequent response.

32. The method of claim 31, wherein the processor is configured to determine the initial
request based on a user document.

33. The method of claim 31, wherein the processor is configured to determine the initial
request based on a keyword selected by the user.

10 34. The method of claim 31, wherein the processor is configured to determine the subsequent
request based on the posting.

35. The method of claim 31, wherein the initial response and the subsequent response
comprise a visual component.

15 36. The method of claim 31, wherein the processor, utilizing a machine-learning module, is
configured to output the subsequent request based on an input of the initial response.

37. The method of claim 36, wherein the processor is configured to determine a category
corresponding to the user.

38. The method of claim 37, wherein the subsequent request is associated with the category.

20 39. The method of claim 31, wherein the video record comprises an audio recording of the
initial response and the subsequent response.

40. The method of claim 31, wherein the processor is configured to determine a score based
on a duration of the initial response and a duration of the subsequent response.

41. An apparatus for matching video records with postings using audiovisual data processing,
the apparatus comprising:

25 at least a processor;
a memory communicatively connected to the at least a processor, the memory
containing instructions configuring the at least a processor to:

receive a posting, the posting comprising a plurality of criteria;

receive a video record;

30 extract a plurality of textual elements from the video record;

- identify a plurality of credentials as a function of the plurality of textual elements;
generate a compatibility score for the video record based on the plurality of criteria and the plurality of credentials; and
5 match the video record with the posting based on the compatibility score.
42. The apparatus of claim 41, wherein the criteria comprise a requirement and a preference.
43. The apparatus of claim 42, wherein the match is based on whether credentials satisfy the requirement.
44. The apparatus of claim 42, wherein the requirement is weighted more than the preference
10 for the compatibility score.
45. The apparatus of claim 41, wherein the video record comprises an audio recording.
46. The apparatus of claim 41, wherein the processor is configured to generate a transcript of the video record.
47. The apparatus of claim 46, wherein the processor is configured to compare the transcript
15 to the posting.
48. The apparatus of claim 41, wherein:
the video record contains visual verbal content; and
the processor is configured to:
use image processing to convert the visual verbal content to textual data;
20 and
compare the textual data to the posting.
49. The apparatus of claim 41, wherein the match is based on an amount of previous matches of the video record.
50. The apparatus of claim 49, wherein the match is based on previous matches of the video
25 record within a specified time.
51. A method for matching video records with postings using audiovisual data processing, the method comprising:
receiving, at a processor, a posting comprising a plurality of criteria;
receiving, at the processor, a video record;
30 extracting, by the processor, a plurality of textual elements from the video record;

identify a plurality of credentials from the video record;

generating, by the processor, a compatibility score for the video record based on the criteria and the credentials; and

matching, by the processor, the video record with the posting based on the compatibility score.

52. The method of claim 51, wherein the criteria comprise a requirement and a preference.

53. The method of claim 52, wherein the match is based on whether credentials satisfy the requirement.

54. The method of claim 52, wherein the requirement is weighted greater than the preference for the compatibility score.

55. The method of claim 51, wherein the video record comprises an audio recording.

56. The method of claim 51, wherein the processor is configured to generate a transcript of the video record.

57. The method of claim 56, wherein the processor is configured to compare the transcript to the posting.

58. The method of claim 51, wherein:
The video record contains visual verbal content; and
the processor is configured to:

use image processing to convert the visual verbal content to textual data; and

compare the textual data to the posting.

59. The method of claim 51, wherein the match is based on an amount of previous matches of the video record.

60. The method of claim 59, wherein the match is based on previous matches of the video record within a specified time.

61. An apparatus for parsing and comparing resume video duplications, the apparatus comprising:

at least a processor; and

a memory communicatively connected to the processor, the memory containing instructions configuring the at least a processor to:

acquire a plurality of video elements from a target video resume, wherein the

target video resume comprises at least an image component;
compare at least an existing video resume and the target video resume to obtain a
comparison result; and
determine, as a function of the comparison result, a duplication coefficient for the
5 target resume video, wherein the duplication coefficient relates to a
similarity between the target video resume and at least an existing resume
video.

62. The apparatus of claim 61, wherein acquiring the plurality of video elements comprises:
identifying a series of frames of the target video resume;

10 identifying a corresponding series of frames of at least an existing video resume;
and
comparing the series of frames of the target video resume to the existing resume
video.

63. The apparatus of claim 62, wherein the target video resume comprises an image
15 reference frame.

64. The apparatus of claim 61, wherein comparing at least an existing video resume and the
target video resume further comprises selecting at least an existing video resume using at
least identifying indicia related to the target video resume.

65. The apparatus of claim 61, wherein the comparison result includes a comparison score
20 that represents a degree of similarity between the target video resume and an existing
video resume of the plurality of existing video resumes.

66. The apparatus of claim 65, wherein the comparison score is determined by dynamic time
warping on the basis of a similarity matrix.

67. The apparatus of claim 61, wherein, as a function of the comparison result, a ranking of
25 the plurality of existing resume videos is generated.

68. The apparatus of claim 67, wherein ranking resume videos comprises linear regression
techniques.

69. The apparatus of claim 68, wherein the linear regression techniques comprise linear
regression models comprising at least ordinary least squares regression.

30 70. The apparatus of claim 61, wherein the duplication coefficient comprises linear

regression analysis.

71. A method of parsing and comparing resume video duplications, using a computing device, the method comprising:

acquiring a plurality of video elements from a target video resume, wherein the target
5 video comprises at least an image component;

comparing, using the computing device, at least an existing video resume and the target
video resume to obtain a comparison result; and

determining, using the computing device, as a function of the comparison result, a
duplication coefficient for the target resume video, wherein the duplication

- 10 coefficient relates to a similarity between the target video resume and at least an
existing resume video.

72. The method of claim 71, wherein acquiring the plurality of video elements comprises:

identifying a series of frames of the target video resume;

identifying a corresponding series of frames of at least an existing video resume;

- 15 and

comparing the series of frames of the target video resume to the existing resume
video.

73. The method of claim 72, wherein the target video resume comprises an image reference
frame.

- 20 74. The method of claim 73, wherein comparing at least an existing video resume and the
target video resume further comprises selecting at least an existing video resume using at
least identifying indica related to the target video resume.

75. The method of claim 71, wherein the comparison result includes a comparison score that
represents a degree of similarity between the target video resume and an existing video
25 resume of the plurality of existing video resumes.

76. The method of claim 71, wherein the comparison score is determined by dynamic time
warping based on a similarity matrix.

77. The method of claim 71, wherein, as a function of the comparison result, a ranking of the
plurality of existing resume videos is generated.

- 30 78. The method of claim 77, wherein ranking video resumes comprises linear regression

techniques.

79. The method of claim 78, wherein the linear regression techniques comprise linear regression models comprising at least ordinary least squares regression.

80. The method of claim 71, wherein the duplication coefficient comprises linear regression analysis.

81. A method for posting match recommendation, the method comprising:
receiving, by a processor, a posting datum of a posting;
receiving, by the processor, a user datum from a user;
retrieving, by the processor, a plurality of records from an immutable sequential listing;
determining, by the processor, a record recommendation as a function of the posting datum, the user datum, and the plurality of records;
providing, by the processor, an interaction preparation as a function of the record recommendation; and
displaying, at a display communicatively connected to the processor, the record recommendation or interaction preparation to a user.

82. The method of claim 81, wherein user datum comprises a draft resume.

83. The method of claim 81, wherein determining the record recommendation comprises generating a compatibility score between the user and the posting.

84. The method of claim 81, wherein the posting datum comprises information related to the posting inputted by the user.

85. The method of claim 81, wherein the user datum comprises information inputted into a graphic user interface of processor by the user.

86. The method of claim 81, wherein the user datum comprises information from a previously presented record.

87. The method of claim 81, storing, by a memory component communicatively connected to the processor, the posting datum.

88. The method of claim 81, wherein the record recommendation comprises a suggested alteration of a record of a user.

89. The method of claim 81, wherein determining the record recommendation comprises generating a user record.

90. The method of claim 81, wherein displaying the interaction preparation comprises displaying the interaction preparation using a graphic user interface.
91. The method of claim 81, wherein the interaction preparation comprises an interactive interview.
- 5 92. A digital posting match recommendation apparatus, the apparatus comprising:
at least a processor; and
a memory communicatively connected to the at least a processor, the memory containing instructions configuring the at least a processor to:
receive a posting datum;
10 determine a record recommendation as a function of the posting datum;
provide an interaction preparation;
determine the record recommendation; and
display, at a display communicatively connected to the at least a processor, the
interaction preparation or record recommendation.
- 15 93. The apparatus of claim 92, wherein the processor is further configured to:
receive, by the processor, a user datum; and
determining, by the processor, a record recommendation as a function of the posting
datum and the user datum.
94. The apparatus of claim 92, wherein the posting datum comprises information from a job
20 posting of a jobseeker.
95. The apparatus of claim 92, wherein the user datum comprises information inputted into a
graphic user interface by the user.
96. The apparatus of claim 92, wherein the user datum comprises information from the
user record.
- 25 97. The apparatus of claim 92, further comprising a memory component that is
communicatively connected to the processor and configured to the posting datum.
98. The apparatus of claim 92, wherein the record recommendation comprises a suggested
alteration of a previously prepared record of a user.
99. The apparatus of claim 92, wherein determining the record recommendation comprises
30 generating a customized record.

100. The apparatus of claim 92, wherein the interaction preparation comprises an interactive interview.
101. An apparatus for linking posting data to a plurality of user identifiers, the apparatus comprising:
- 5 at least a processor; and
- a memory communicatively connected to the processor, the memory containing instructions configuring the at least a processor to:
- receive a plurality of user identifiers relating to a plurality of users, wherein a user identifier of the plurality of user identifiers comprises at least a video
- 10 resume;
- receive posting data from a posting generator of a plurality of posting generators;
- identify a plurality of keywords within the posting data;
- generate a keyword ranking based on a plurality of weighted values reflecting priority among the plurality of keywords;
- 15 match a plurality of keywords of the keyword ranking to the plurality of user identifiers; and
- generate, as a function of the matching, a ranking of the plurality of user identifiers based on a superiority criterion of each user identifier using a fuzzy set inference system.
- 20 102. The apparatus of claim 101, wherein:
- the at least video resume contains at least verbal content; and
- the memory contains instructions further configuring the processor to utilize at least an audiovisual speech recognition process to convert the verbal content into textual data.
- 25 103. The apparatus of claim 101, wherein the posting data comprises a plurality of descriptive data elements pertaining to the posting generator.
104. The apparatus of claim 101, wherein the plurality of keywords comprises at least the plurality of descriptive data elements.
105. The apparatus of claim 101, wherein the memory contains instructions further
- 30 configuring the processor to use a language processing module to extract the plurality of

keywords from the posting data.

106. The apparatus of claim 101, wherein the memory contains instructions further configuring the processor to use optical character recognition to extract the plurality of keywords from the posting data.

5 107. The apparatus of claim 101, wherein generating the keyword ranking based on the plurality of weighted values comprises using a machine learning process including a feedback loop to generate the plurality of weighted values based on training data containing at least priority feedback from the plurality of posting generators.

10 108. The apparatus of claim 107, wherein the at least priority feedback from the plurality of posting generators is received from a priority feedback database containing a plurality of samples related to the keyword ranking.

109. The apparatus of claim 101, wherein matching a plurality of keywords to the plurality of user identifiers comprises using a machine learning classification algorithm configured to output data bins of a plurality of particular user identifiers matched to the posting data.

15 110. The apparatus of claim 109, wherein using the machine learning classification algorithm further comprises using a language processing module.

111. A method for using linking posting data to a plurality of user identifiers, the method comprising:

20 receiving, using a computing device, a plurality of identifiers relating to a plurality of users, wherein a user identifier of the plurality of user identifiers comprises at least a video resume;

receiving, using a computing device, posting data from a posting generator of a plurality of posting generators;

25 identifying, using a computing device, a plurality of keywords within the received posting data;

generating, using a computing device, a keyword ranking based on a plurality of weighted values reflecting priority among the plurality of keywords;

matching, using a computing device, a plurality of keywords of the keyword ranking to the plurality of user identifiers; and

30 generating, using a computing device, a ranking of the plurality of user identifiers based

on a superiority criterion of each user identifier using a fuzzy set inference system.

112. The method of claim 111, wherein:

the at least video resume contains at least verbal content; and

5 the method further comprises using at least an audiovisual speech recognition process to convert the verbal content into textual data.

113. The method of claim 111, wherein the posting data comprises a plurality of descriptive data elements pertaining to the posting generator.

10 114. The method of claim 111, wherein the plurality of keywords comprises at least the plurality of descriptive data elements.

115. The method of claim 111, wherein the method further comprises using language processing module to extract a plurality of keywords from the received posting data.

116. The apparatus of claim 111, wherein the method includes using optical character recognition to extract the plurality of keywords from the posting data.

15 117. The method of claim 111, wherein generating the keyword ranking based on the plurality of weighted values comprises using a machine learning process including a feedback loop to generate the plurality of weighted values based on training data containing at least priority feedback from the plurality of posting generators.

20 118. The apparatus of claim 117, wherein the at least priority feedback from the plurality of posting generators is received from a priority feedback database containing a plurality of samples related to the keyword ranking.

119. The method of claim 111, wherein matching a plurality of keywords to the plurality of user identifiers comprises using a machine learning classification algorithm configured to output data bins of a plurality of particular user identifiers matched to the posting data.

25 120. The method of claim 119, wherein using the machine learning classification algorithm further comprises using a language processing module.

121. An apparatus for linking posting data to a plurality of user identifiers, the apparatus comprising:

at least a processor; and

30 a memory communicatively connected to the processor, the memory containing

instructions configuring the at least a processor to:

receive a plurality of user identifiers relating to a plurality of users, wherein a user identifier of the plurality of user identifiers comprises at least a video resume;

5 receive posting data from a posting generator of a plurality of posting generators; identify a plurality of keywords within the posting data;

generate a keyword ranking, wherein generating a keyword list ranking comprises generating, by the at least a processor, a plurality of weighted values as a function of training data containing at least priority feedback from the plurality of posting generators as a function of a machine learning process containing a feedback loop, wherein the plurality of weighted values reflect priority among the plurality of keywords;

10 match a plurality of keywords of the keyword ranking to the plurality of user identifiers; and

15 generate, as a function of the matching, a ranking of the plurality of user identifiers based on a superiority criterion of each user identifier using a fuzzy set inference system.

122. An apparatus for matching video records with postings using audiovisual data processing, the apparatus comprising:

20 at least a processor;

a memory communicatively connected to the at least a processor, the memory containing instructions configuring the at least a processor to:

receive a posting, the posting comprising a plurality of criteria;

receive a video record comprising at least visual verbal content;

25 extract a plurality of textual elements from the video record;

identify a plurality of credentials as a function of the plurality of textual elements by utilizing optical character recognition to recognize at least a keyword as a function of the visual verbal content;

30 generate a compatibility score for the video record based on the plurality of criteria and the plurality of credentials; and

match the video record with the posting based on the compatibility score.

123. An apparatus for parsing and comparing resume video duplications, the apparatus comprising:

at least a processor; and

5 a memory communicatively connected to the processor, the memory containing instructions configuring the at least a processor to:

decode a target video resume wherein decoding the target video resume further comprises identifying a plurality of reference frames;

10 parse a plurality of reference frames of the target video resume for a plurality of video elements;

acquire the plurality of video elements from the target video resume, wherein the target video resume comprises at least an image component;

compare at least an existing video resume and the target video resume to obtain a comparison result;

15 determine, as a function of the comparison result, a duplication coefficient for the target resume video, wherein the duplication coefficient relates to a similarity between the target video resume and at least an existing resume video; and

20 flagging, as a function of the duplication coefficient and the at least an existing resume video, the target resume video.

124. A method for posting match recommendation, the method comprising:

receiving, by a processor, a posting datum of a posting;

receiving, by the processor, a user datum from a user;

25 retrieving, by the processor, a plurality of user records from an immutable sequential listing, wherein:

the immutable sequence listing includes a plurality of resume data of the user and a plurality of cryptographically secured sub-listings such that each of the plurality of cryptographically secured sub-listings is linked to a previous sub-listing in chronological order; and

30 the processor is configured to verify content posted in the immutable sequential

listing when the processor traverses the plurality of cryptographically
secured sub-listings in reverse chronological order;
determining, by the processor, a record recommendation as a function of the posting
datum, the user datum, and the plurality of user records, wherein determining the
5 record recommendation further comprises:
training a machine-learning model using training data, wherein the training data
correlates user data to posting data; and
generating, using the trained machine-learning model, the record
10 recommendation, wherein at least a portion of the plurality of resume data
is provided to the trained machine-learning as an input to output the record
recommendation;
providing, by the processor, an interaction preparation as a function of the record
recommendation; and
displaying, at a display communicatively connected to the processor, the record
15 recommendation or the interaction preparation to the user.

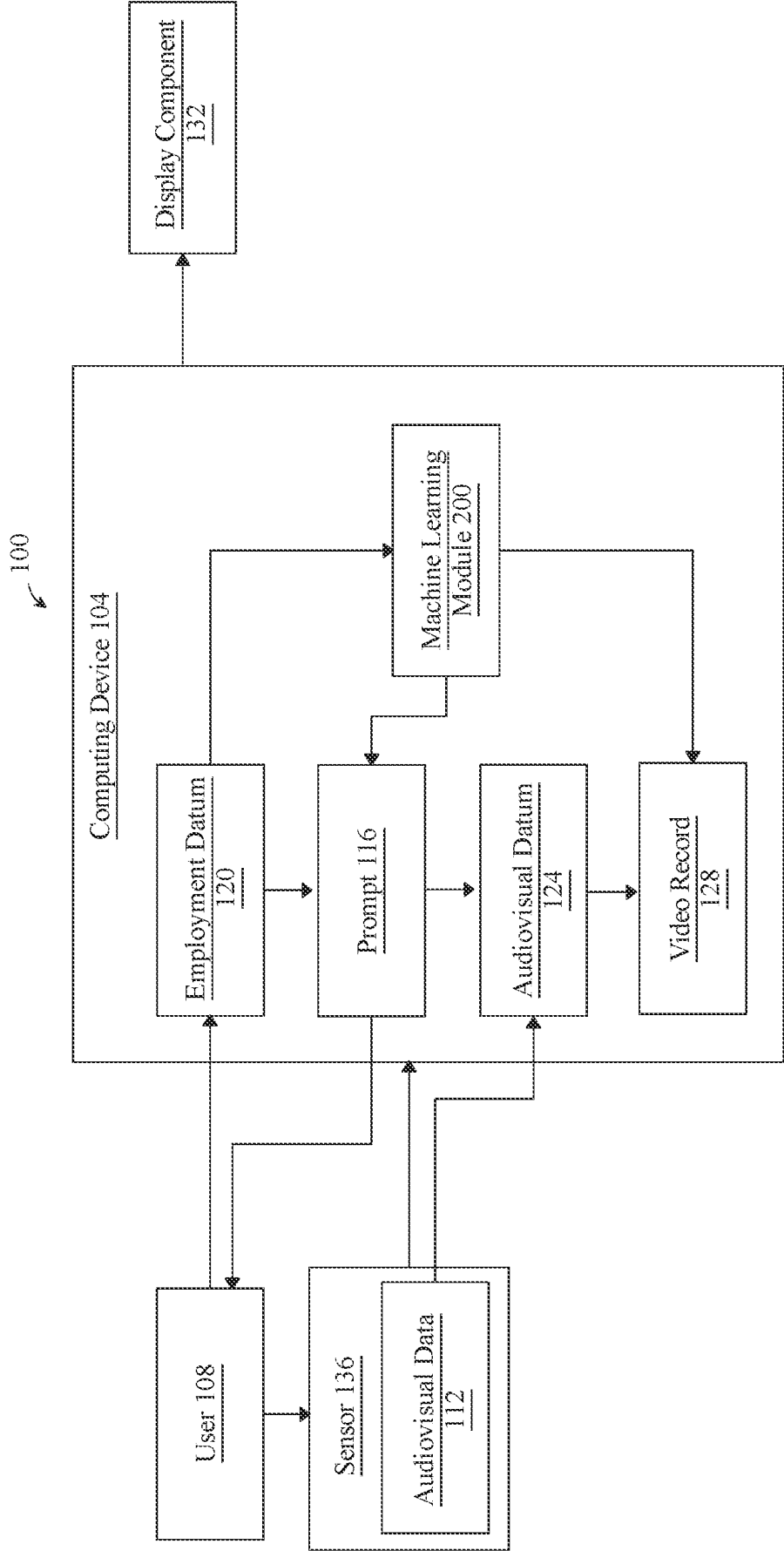


FIG. 1

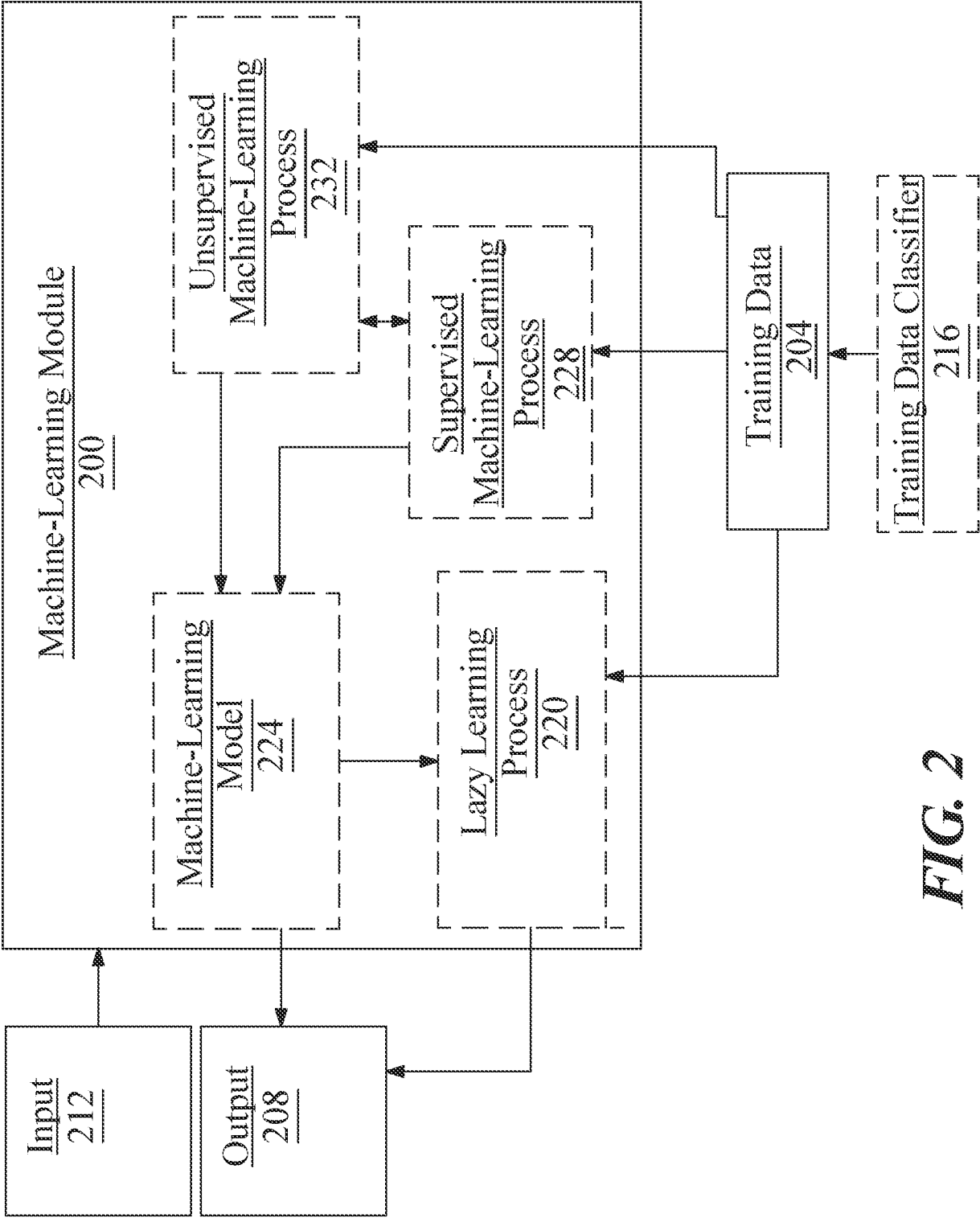
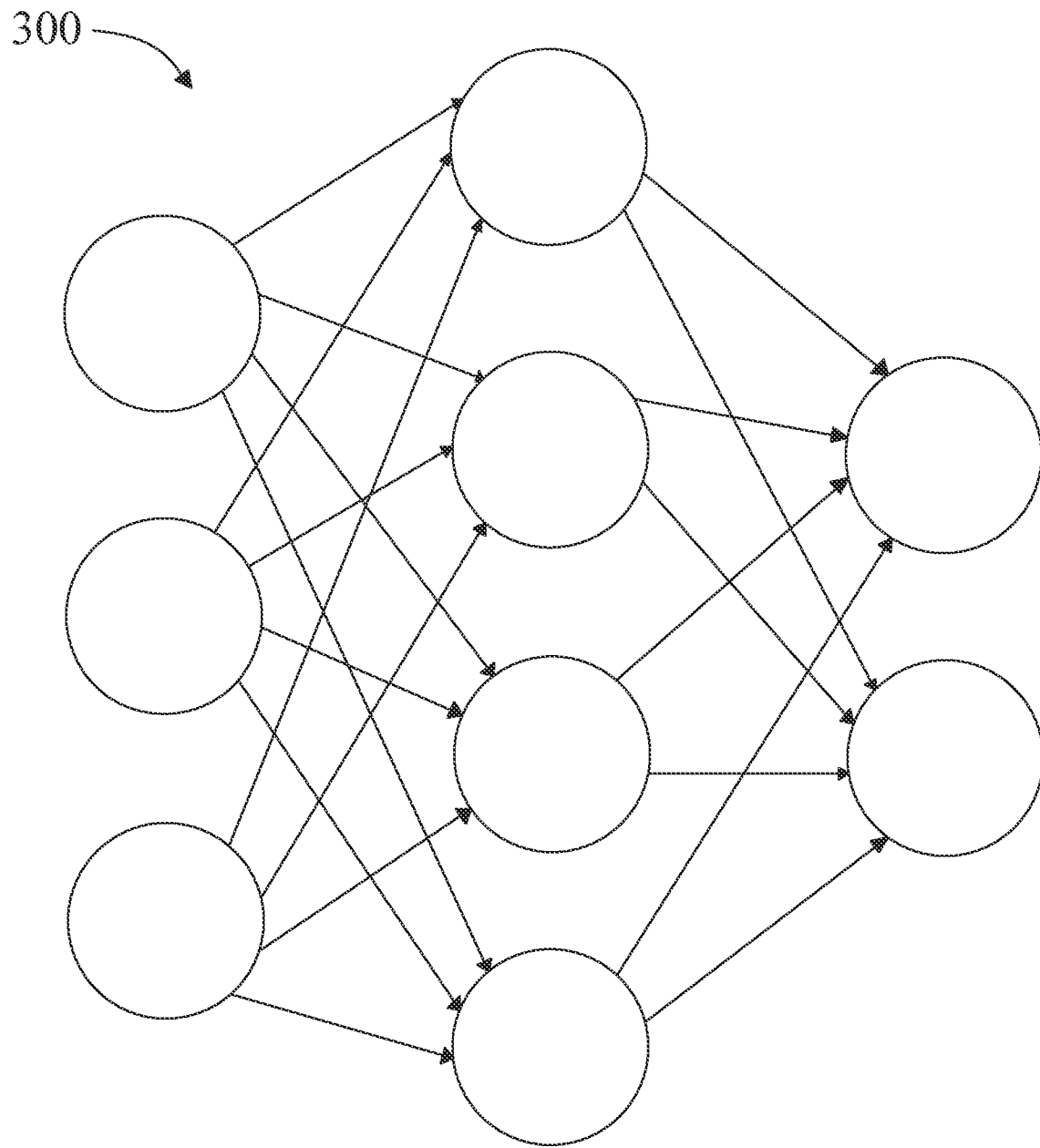


FIG. 2

***FIG. 3***

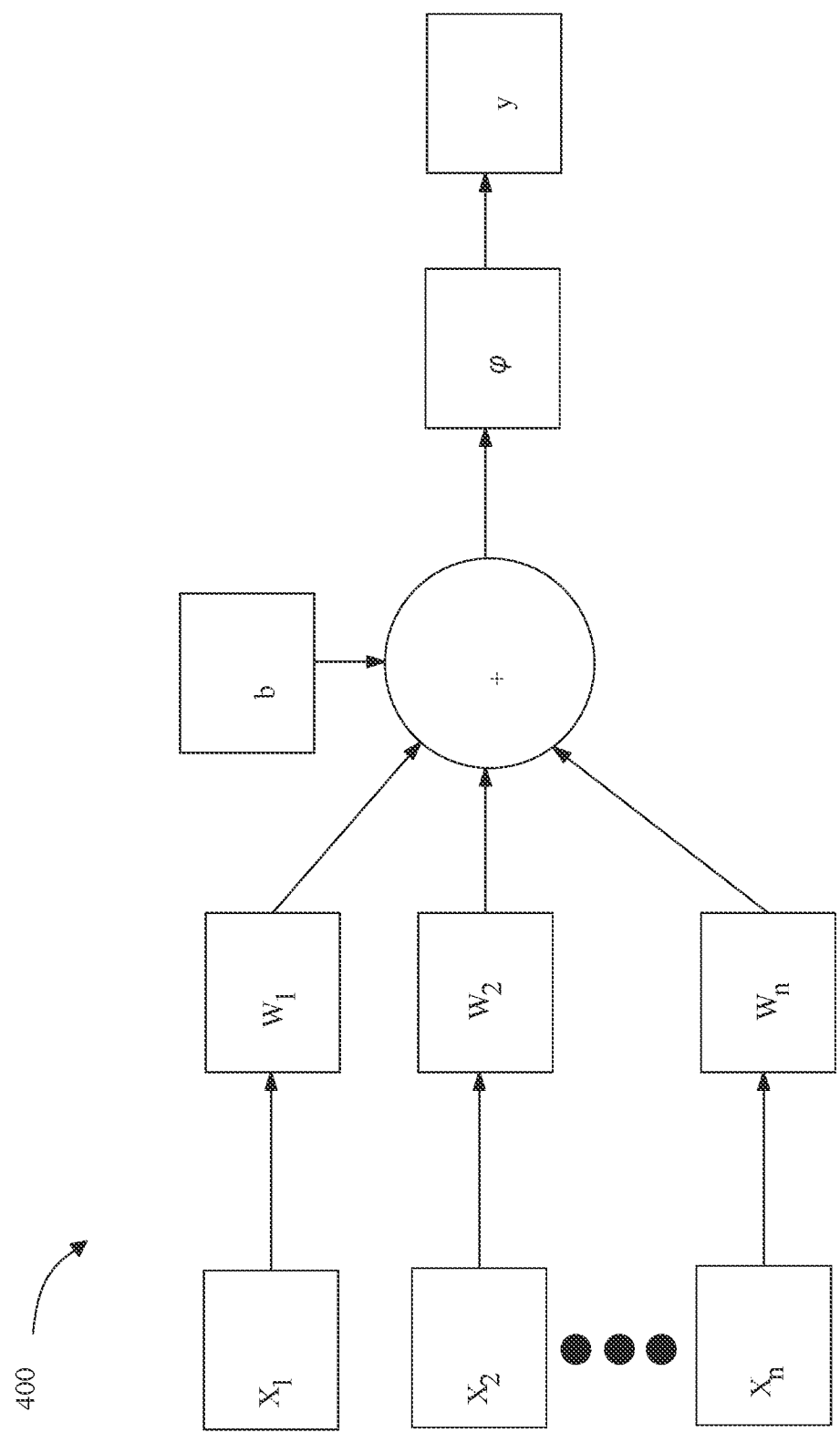


FIG. 4

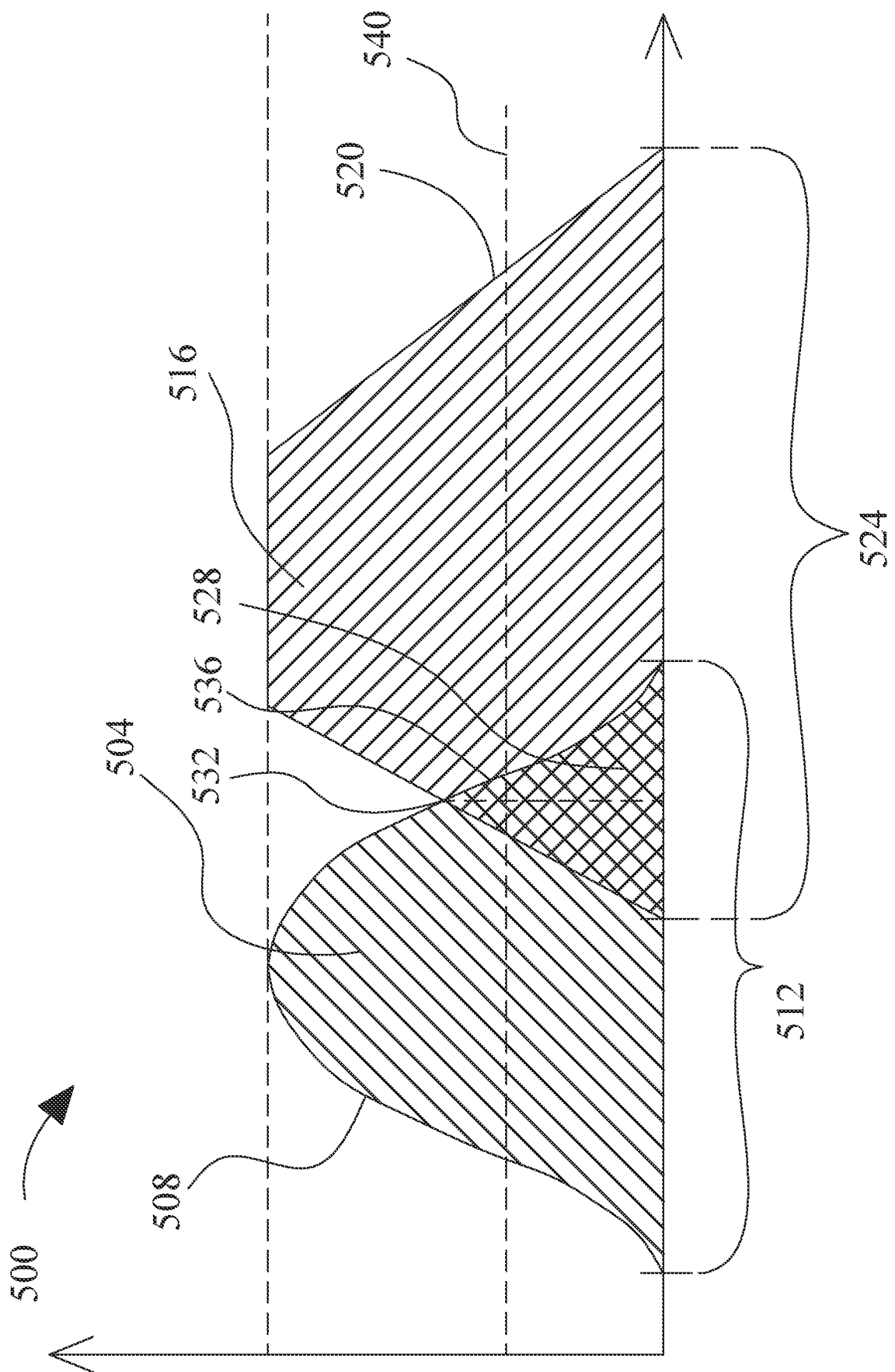
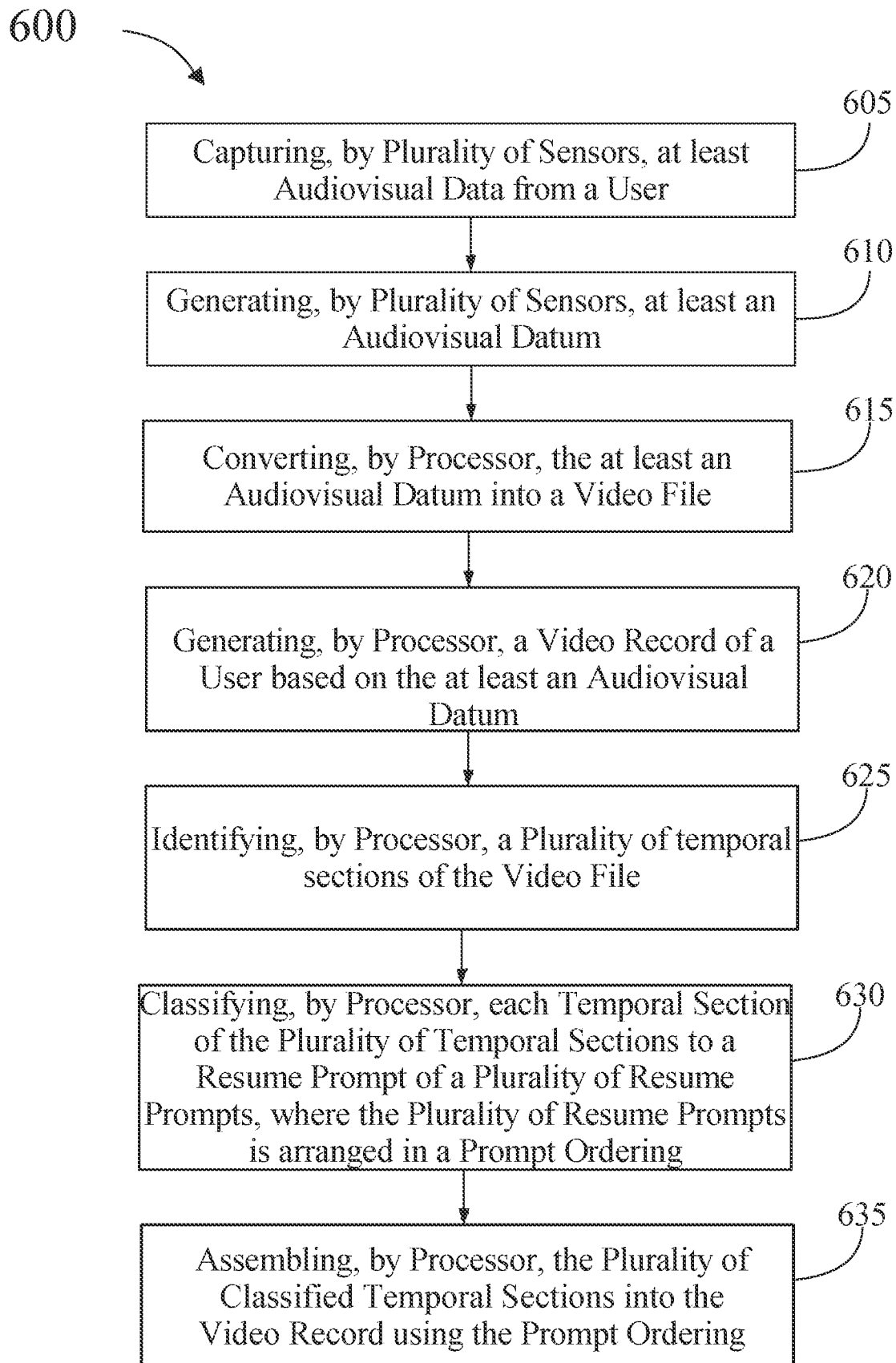


FIG. 5

**FIG. 6**

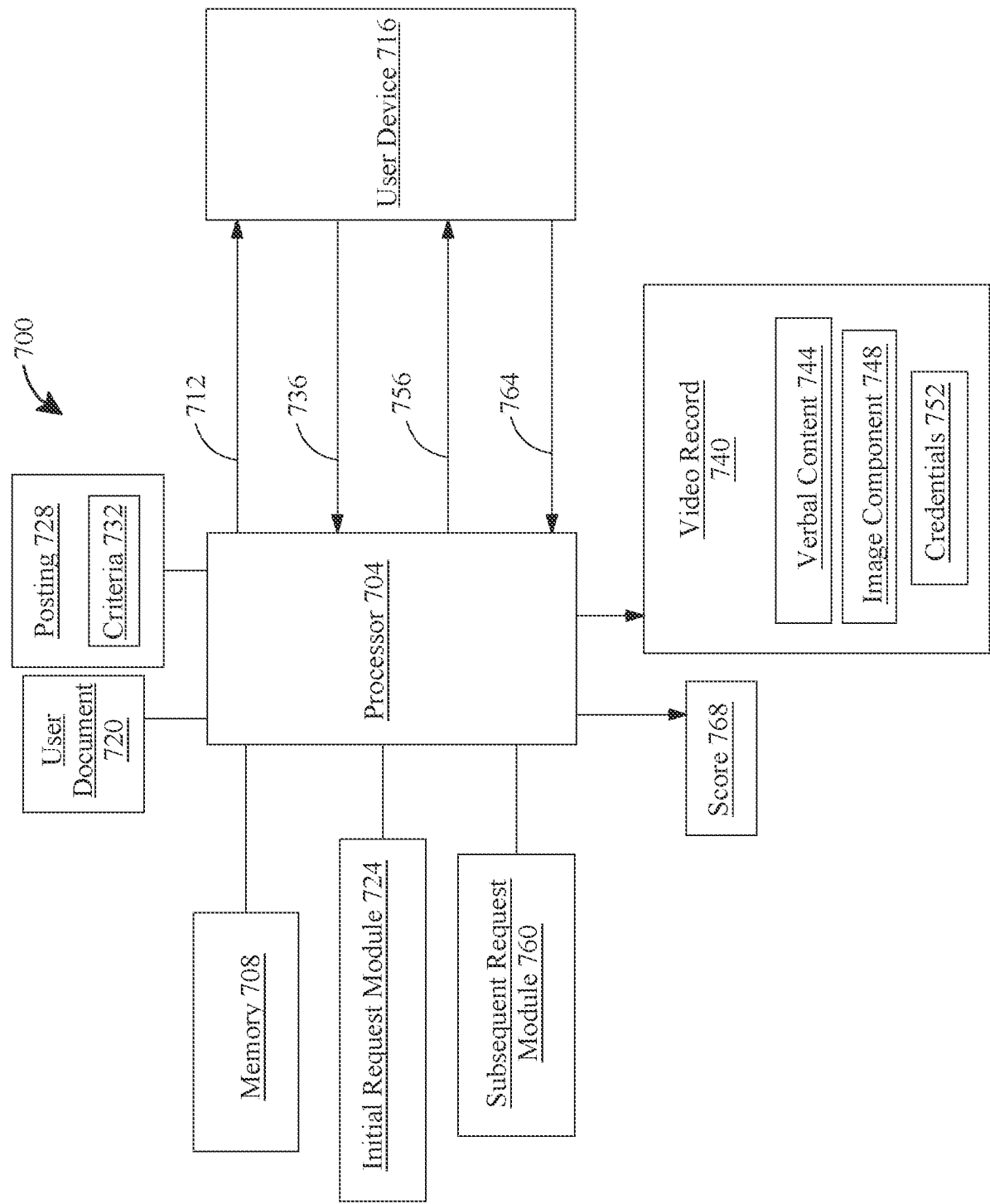


FIG. 7

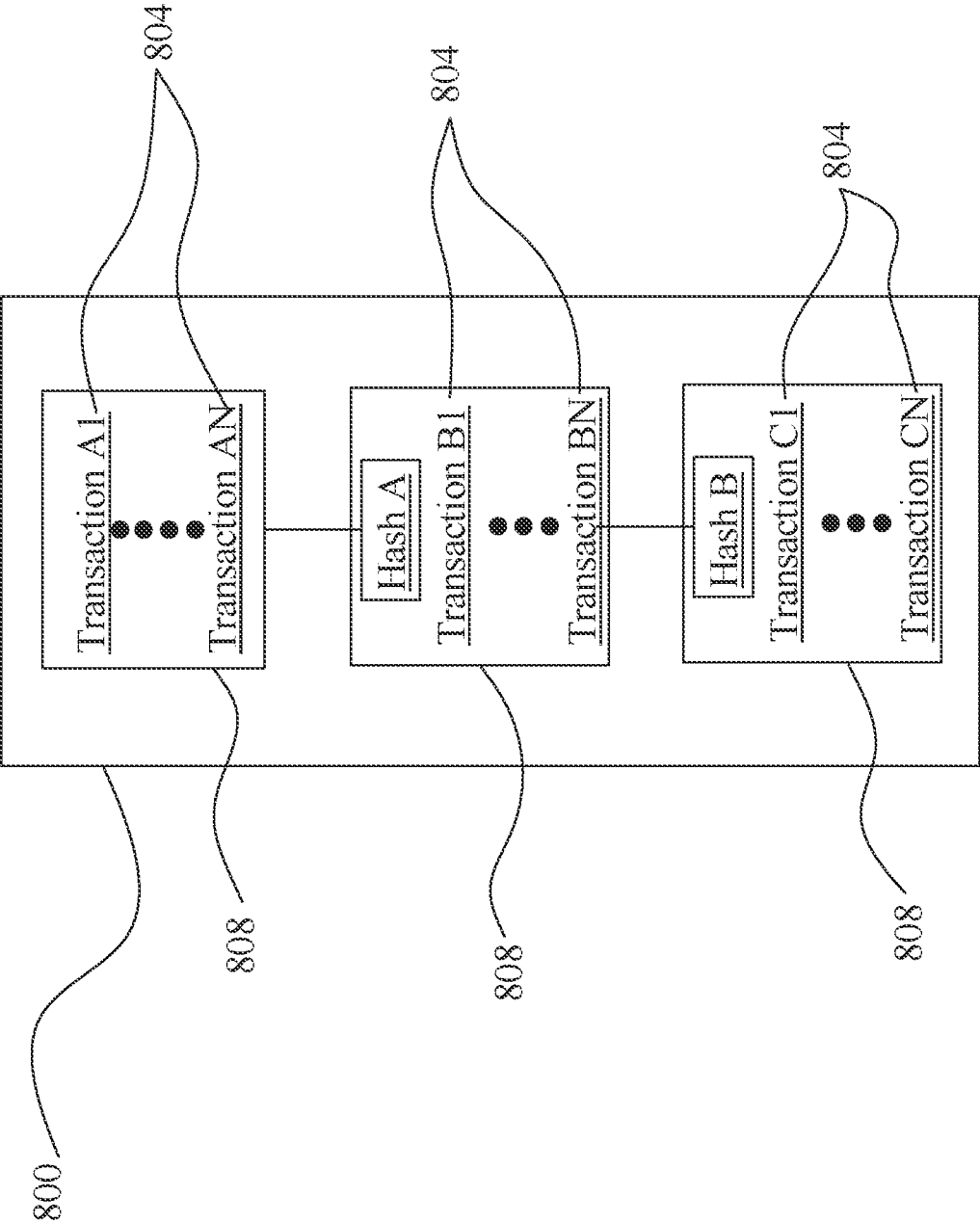
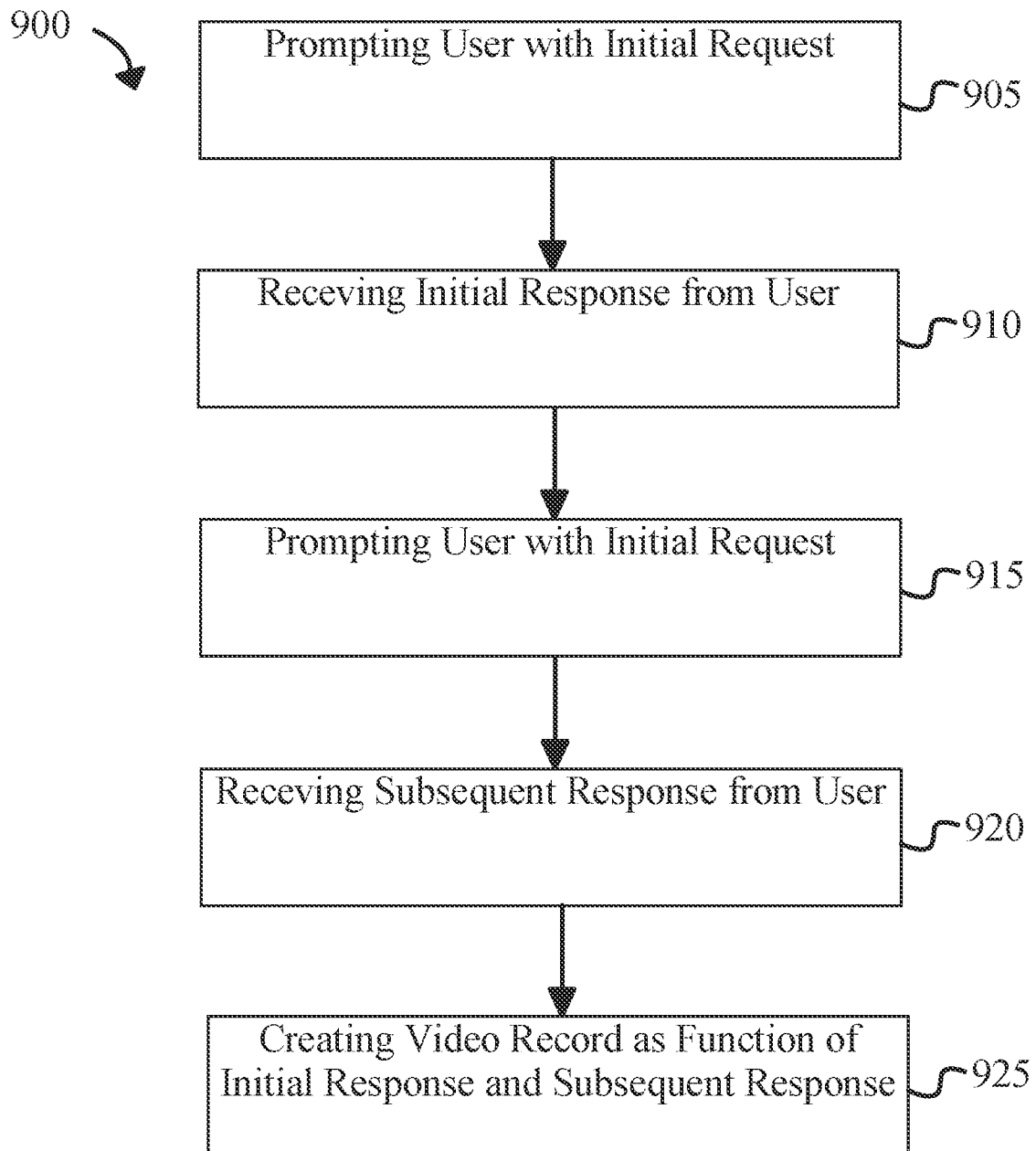


FIG. 8

**FIG. 9**

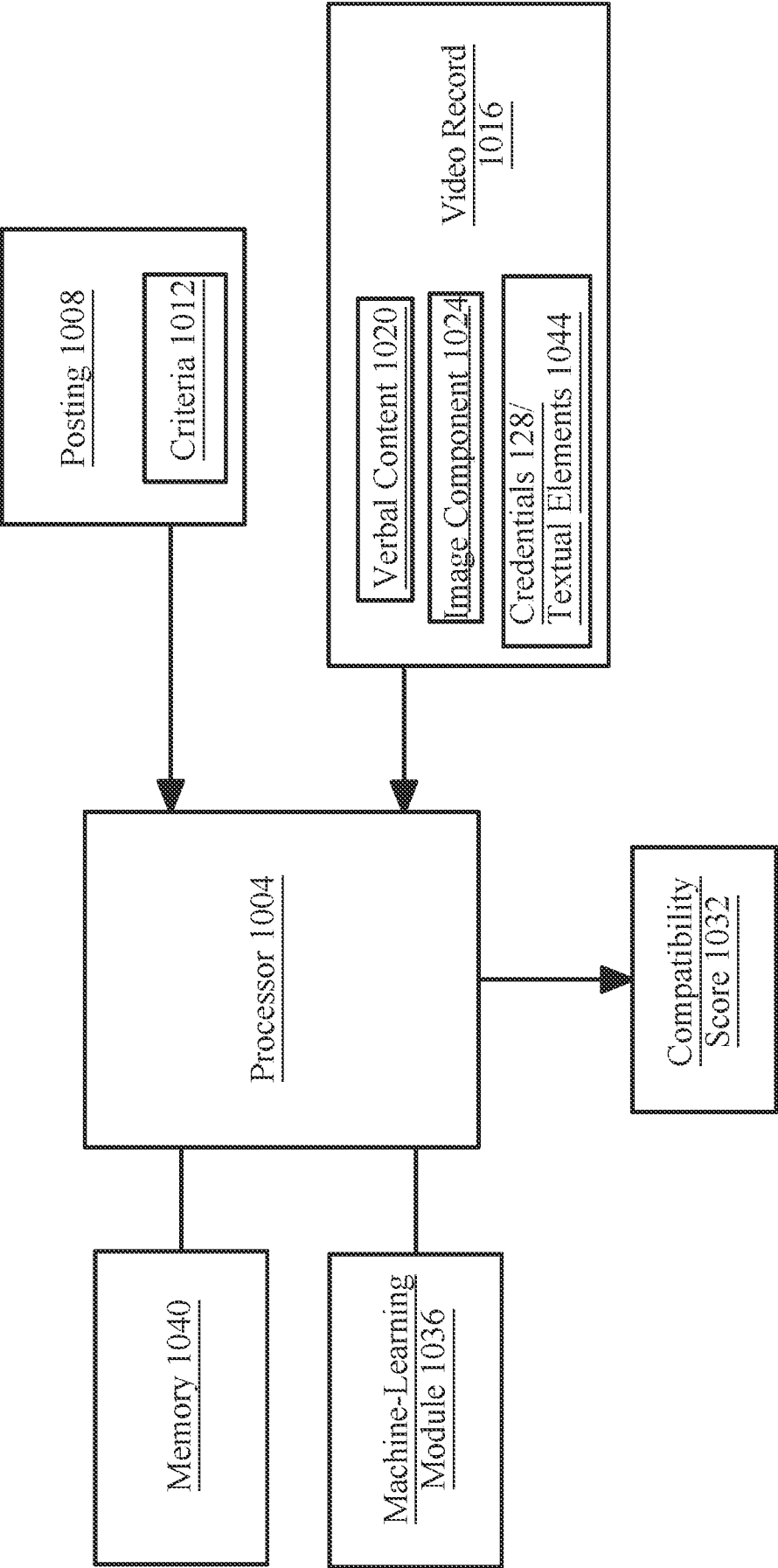
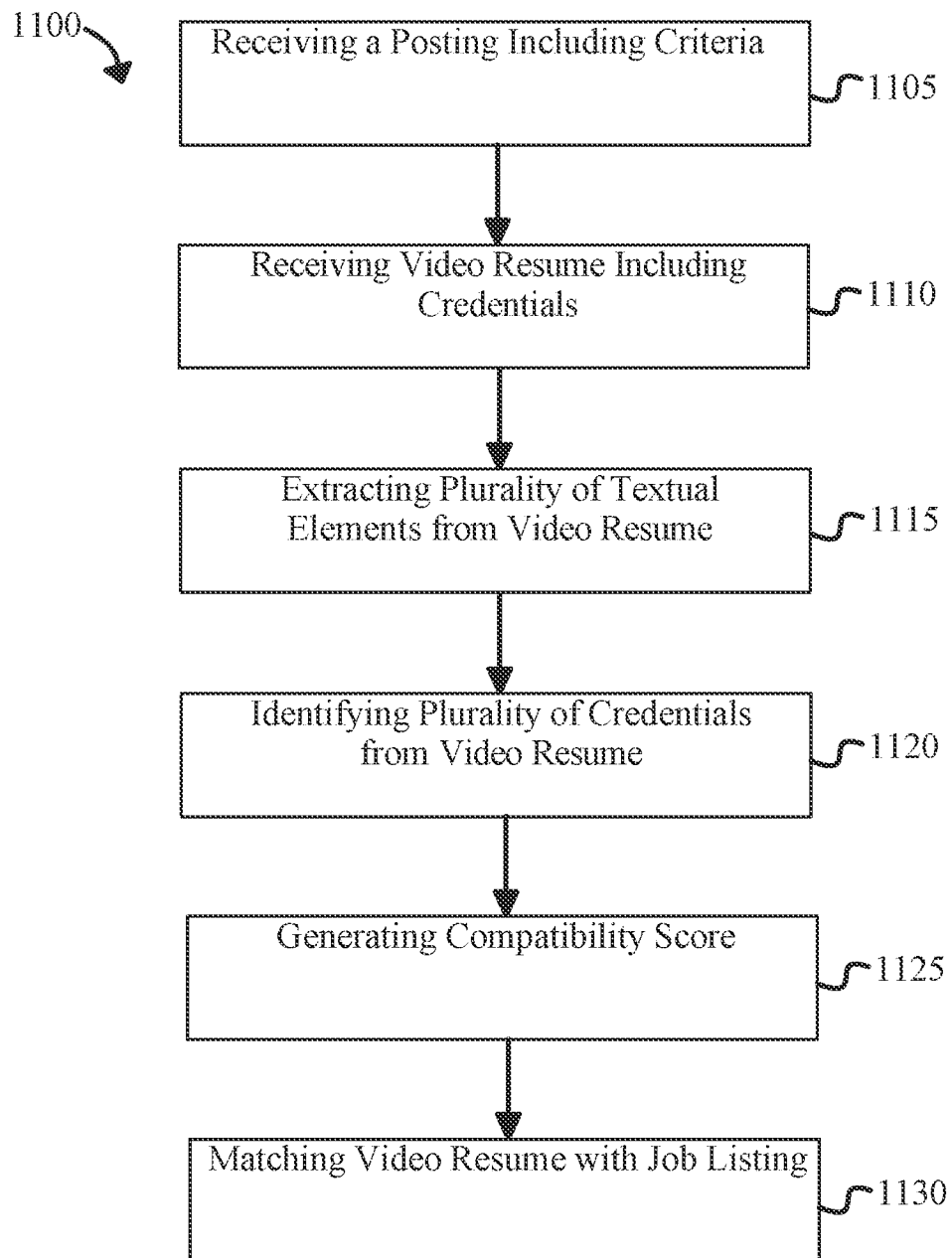


FIG. 10

**FIG. 11**

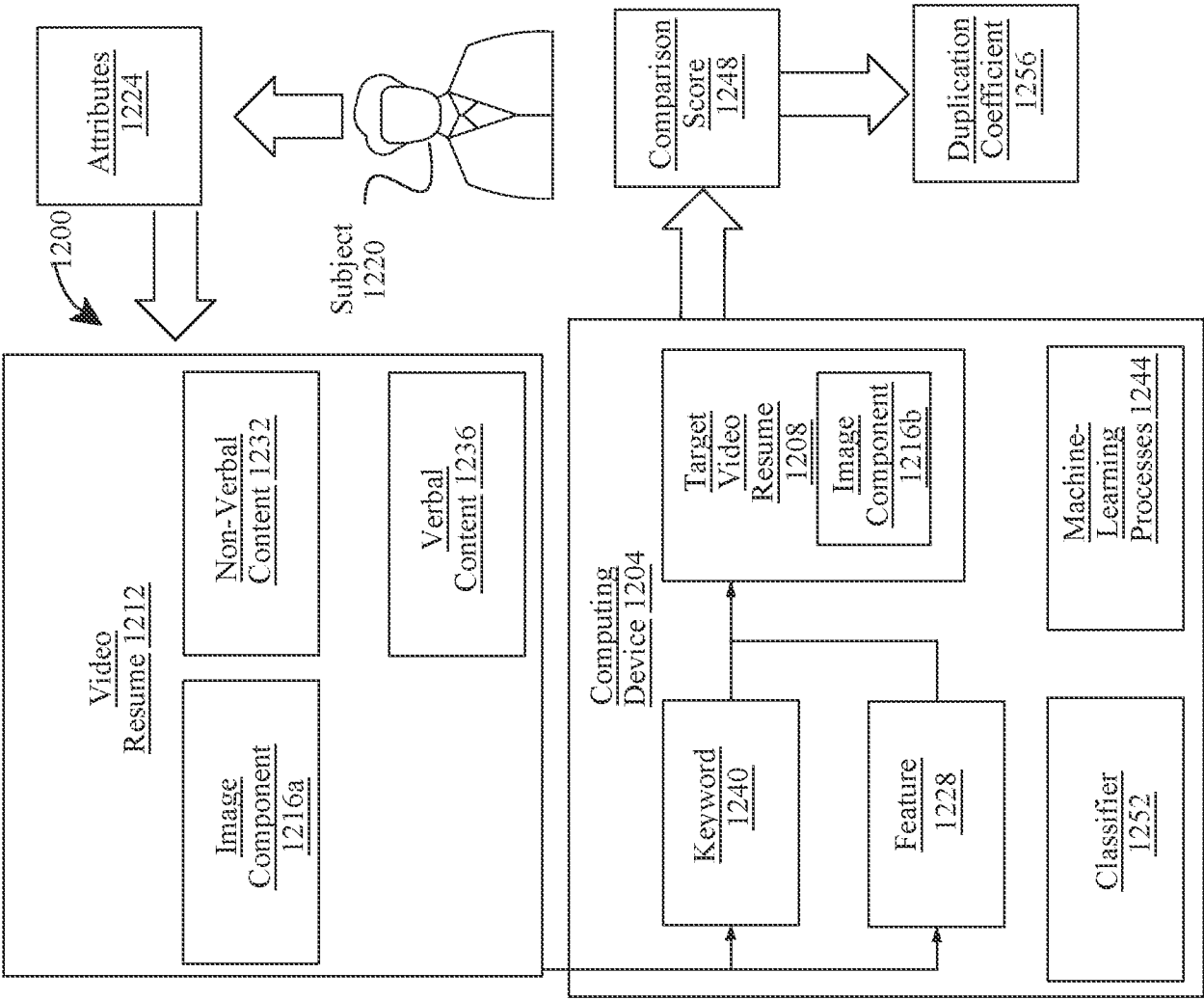
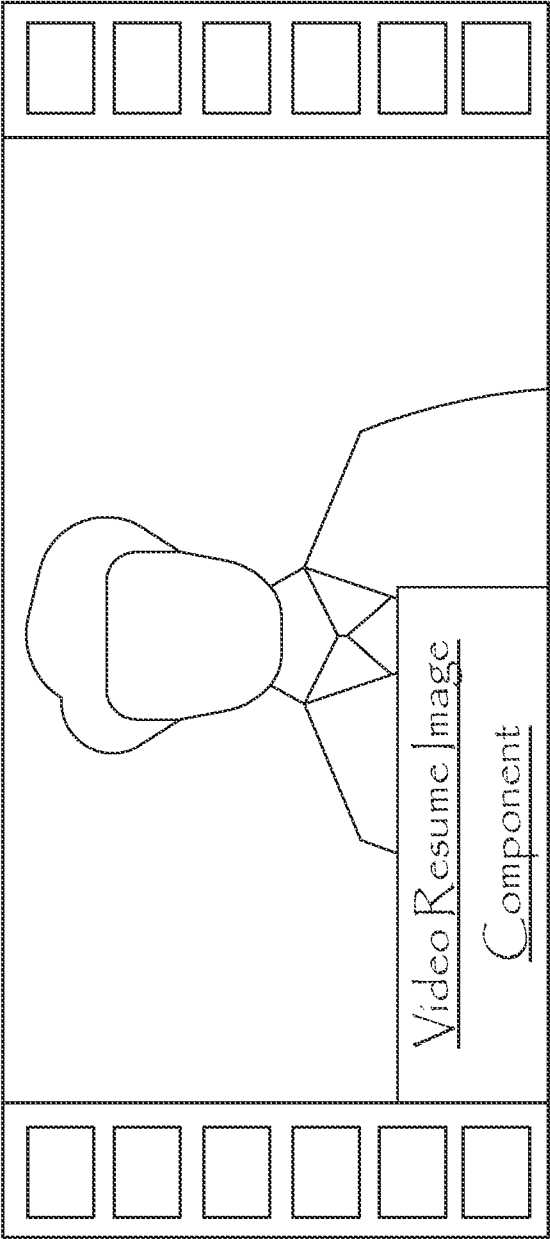


FIG. 12



1300

FIG. 13

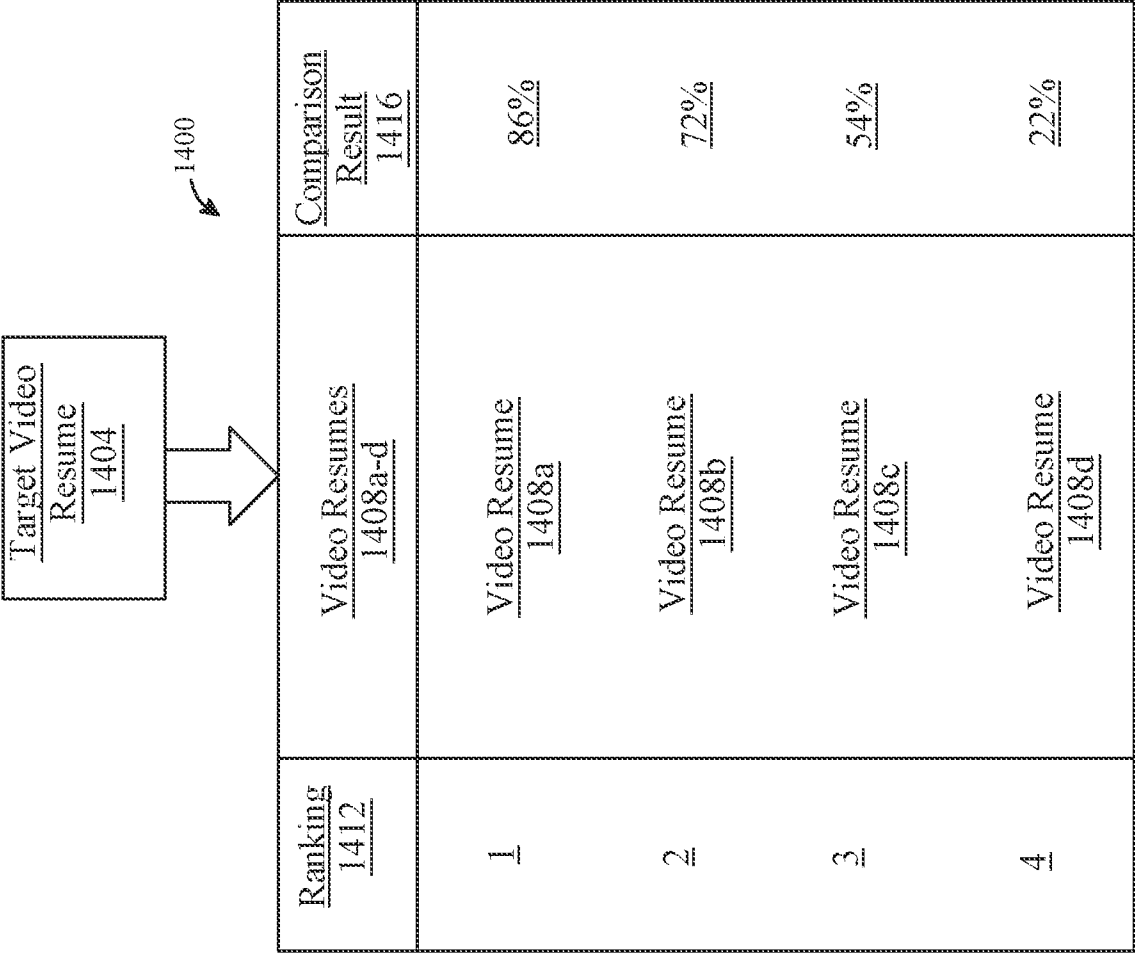


FIG. 14

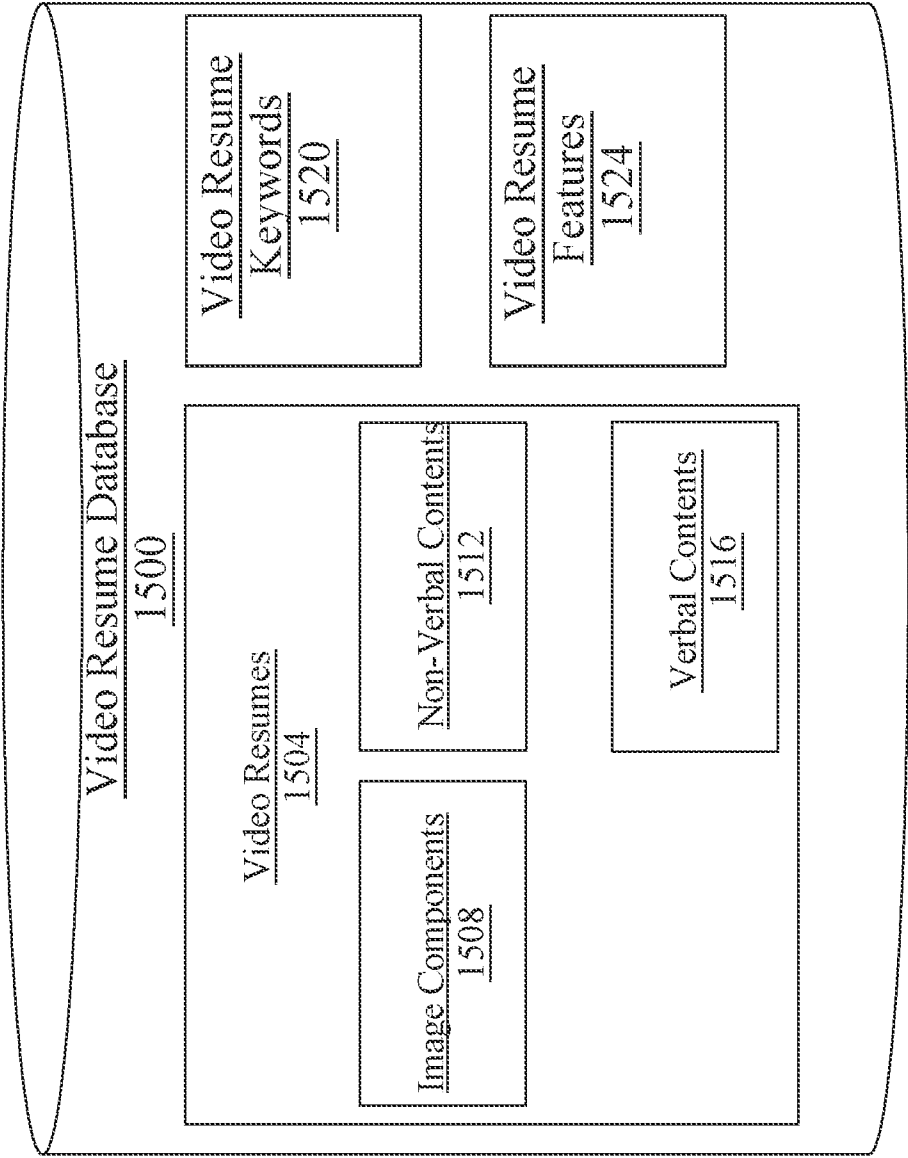


FIG. 15

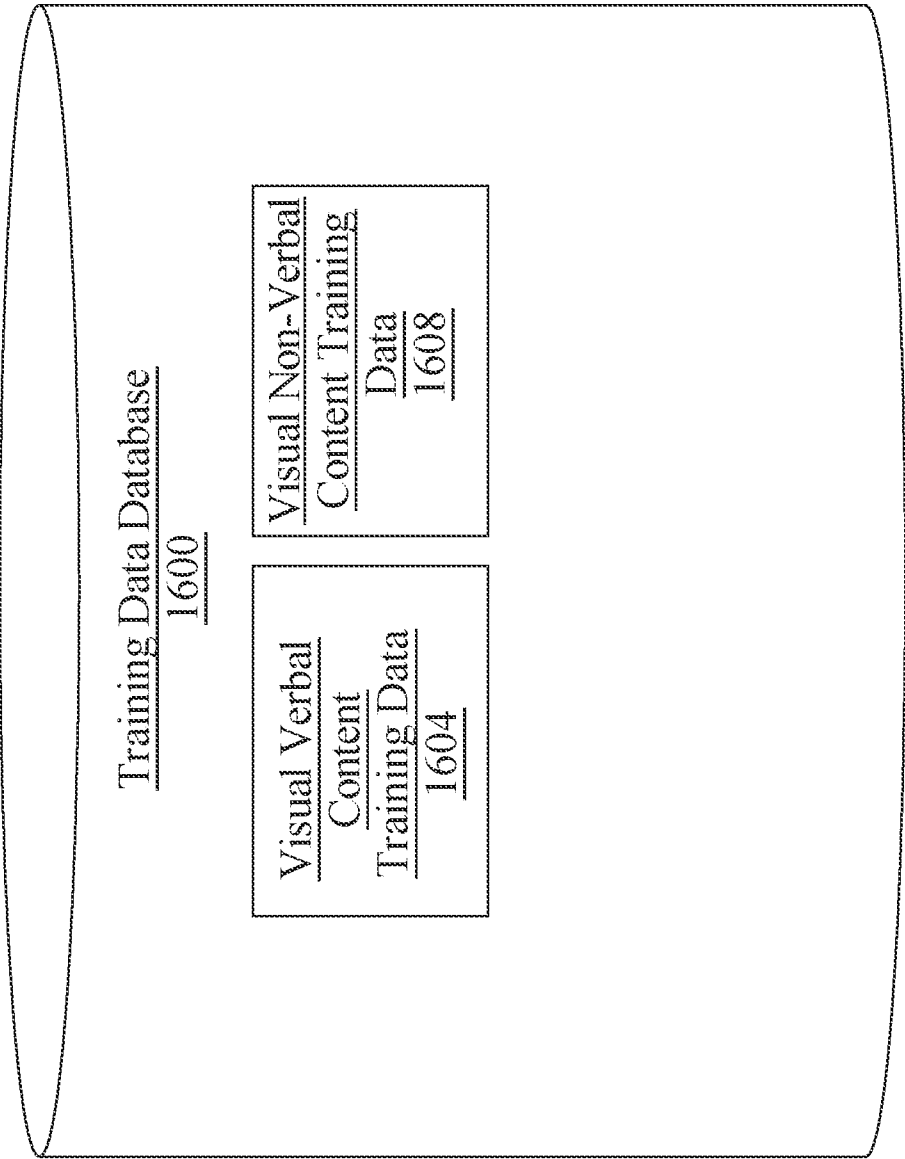
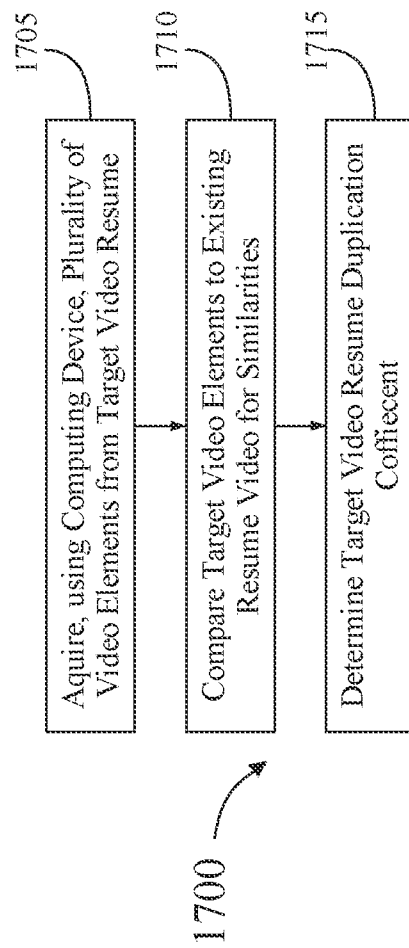


FIG. 16

**FIG. 17**

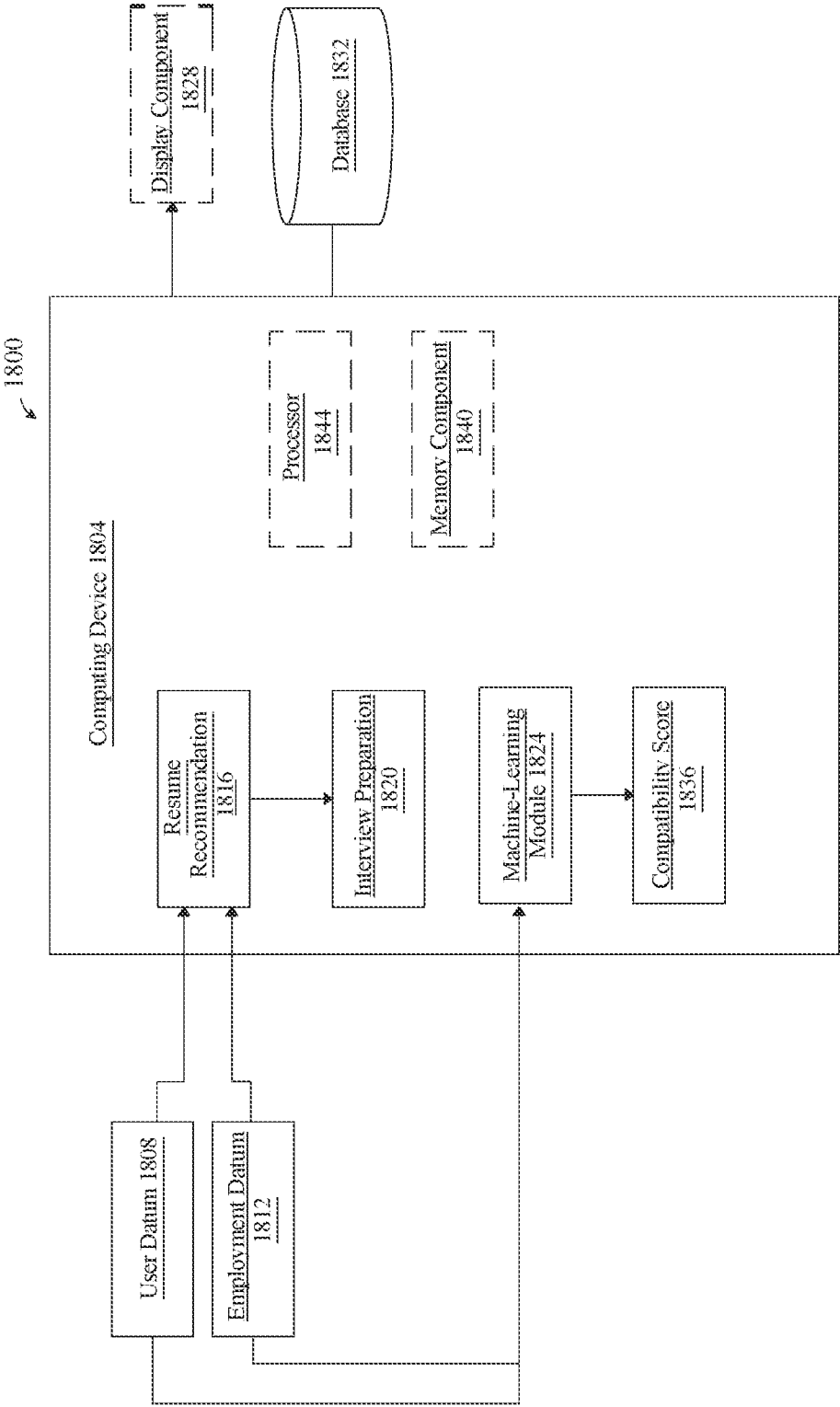
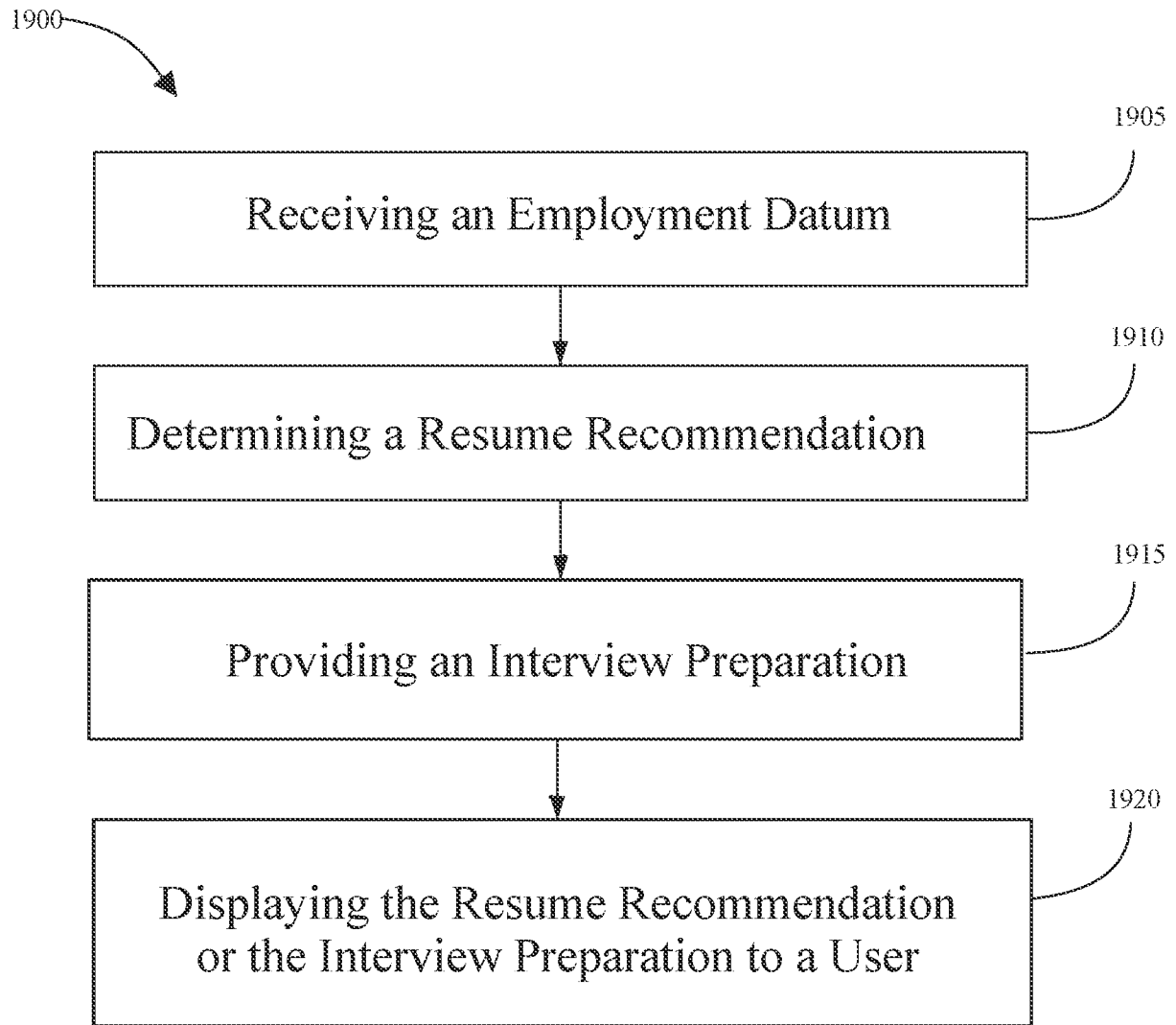


FIG. 18

**FIG. 19**

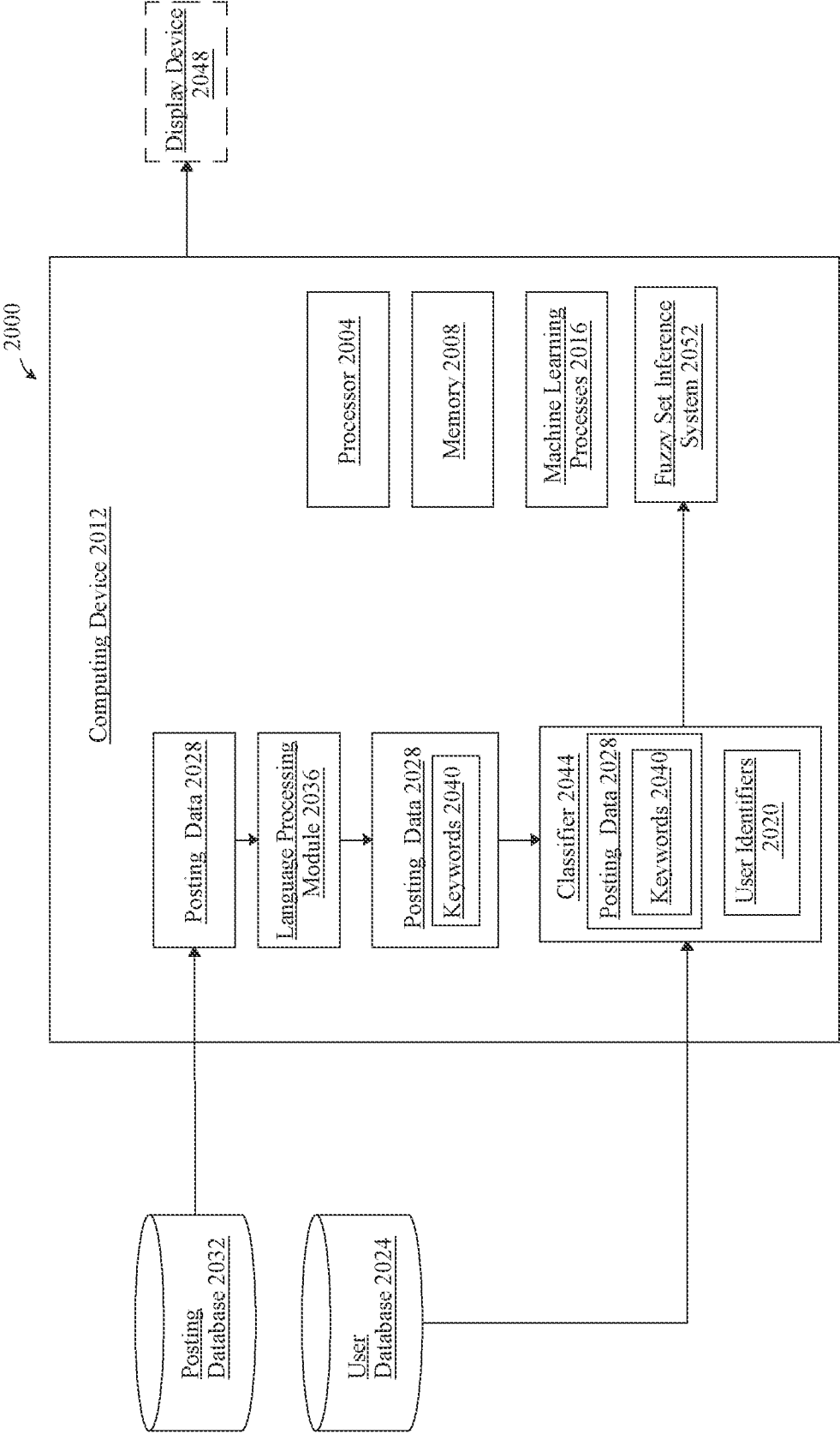


FIG. 20

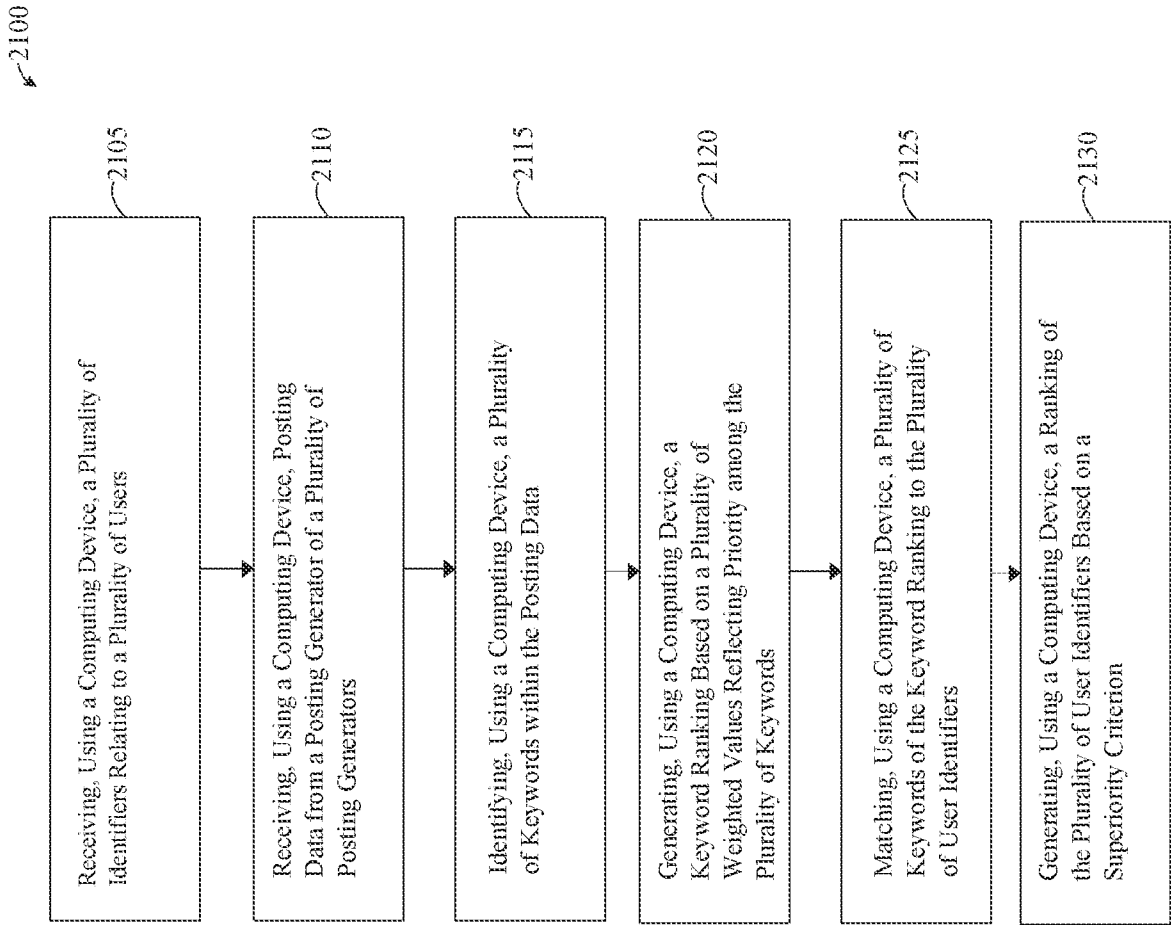
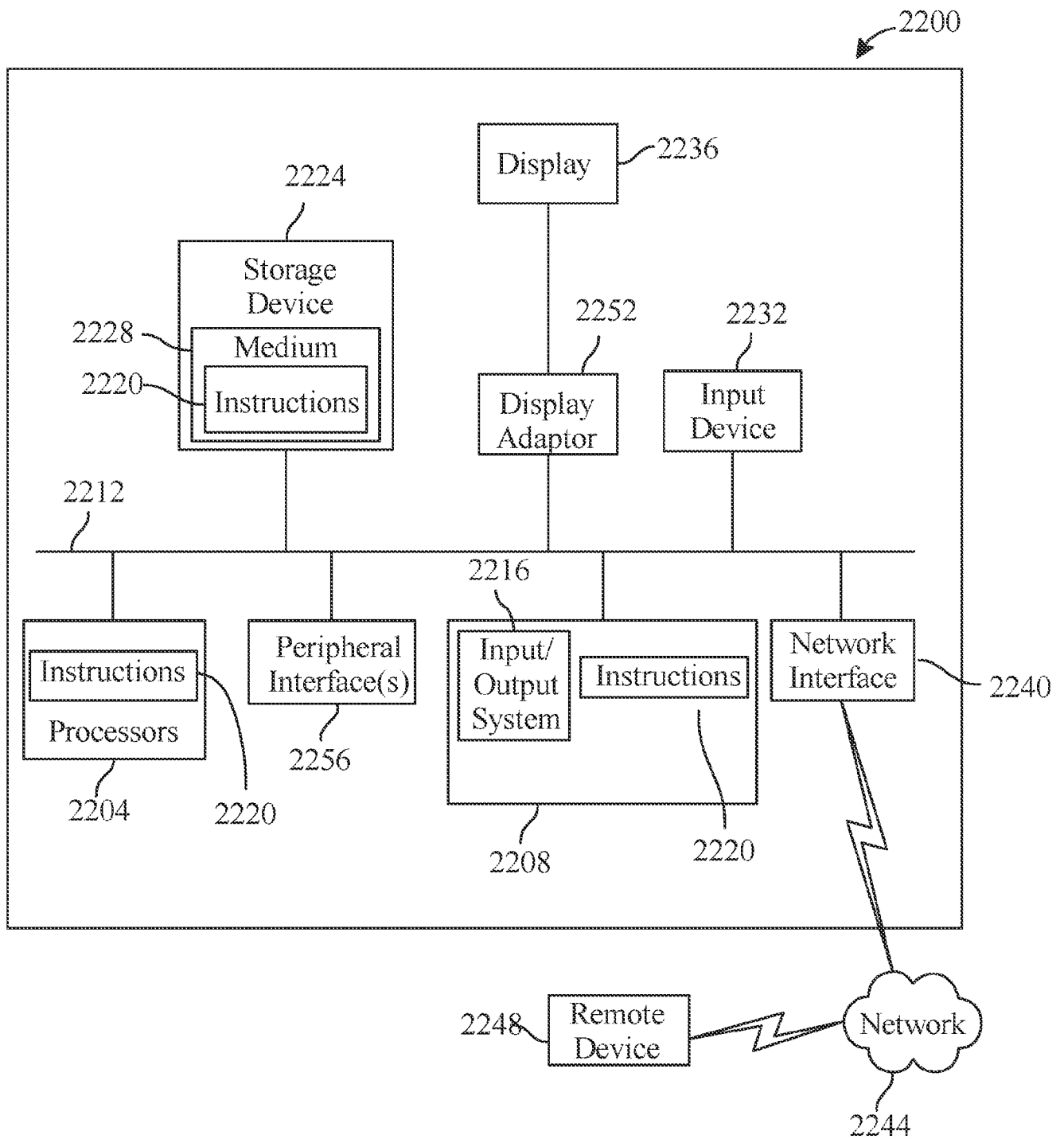


FIG. 21

**FIG. 22**