



(86) Date de dépôt PCT/PCT Filing Date: 2010/06/01

(87) Date publication PCT/PCT Publication Date: 2010/12/09

(45) Date de délivrance/Issue Date: 2017/04/25

(85) Entrée phase nationale/National Entry: 2011/11/03

(86) N° demande PCT/PCT Application No.: US 2010/036964

(87) N° publication PCT/PCT Publication No.: 2010/141513

(30) Priorité/Priority: 2009/06/04 (US12/477,918)

(51) Cl.Int./Int.Cl. *G06F 17/00* (2006.01),
G06F 13/14 (2006.01), *G06F 15/16* (2006.01),
G06F 17/18 (2006.01), *G06F 17/27* (2006.01)

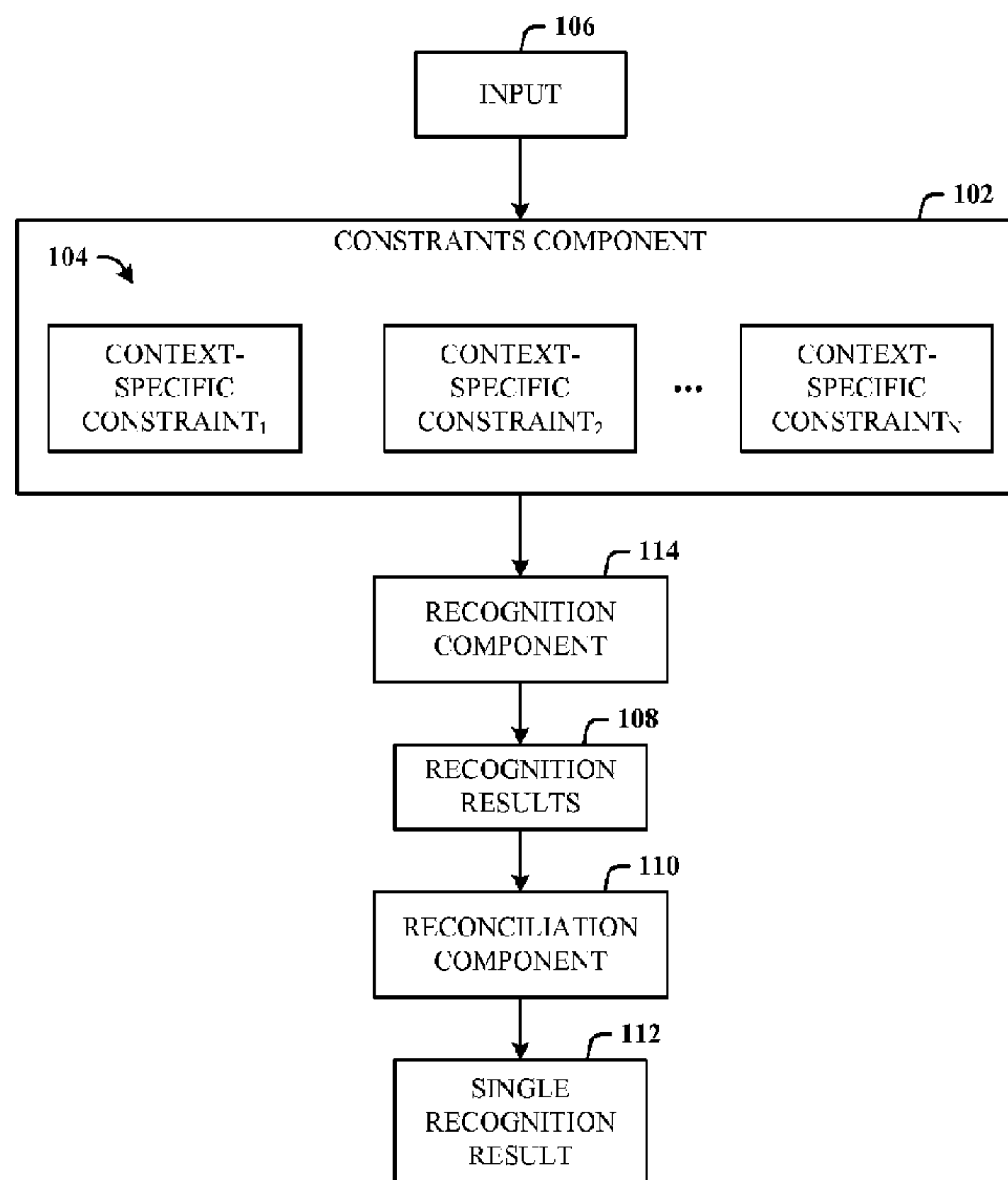
(72) Inventeurs/Inventors:
CHANG, SHUANGYU, US;
LEVIT, MICHAEL, US;
BUNTSCHUH, BRUCE, US

(73) Propriétaire/Owner:
MICROSOFT TECHNOLOGY LICENSING, LLC, US

(74) Agent: SMART & BIGGAR

(54) Titre : RECONNAISSANCE A L'AIDE D'UNE NOUVELLE RECONNAISSANCE ET D'UNE CLASSIFICATION
STATISTIQUE

(54) Title: RECOGNITION USING RE-RECOGNITION AND STATISTICAL CLASSIFICATION



(57) **Abrégé/Abstract:**

Architecture that employs an overall grammar as a set of context-specific grammars for recognition of an input, each responsible for a specific context, such as subtask category, geographic region, etc. The grammars together cover the entire domain. Moreover, multiple recognitions can be run in parallel against the same input, where each recognition uses one or more of the context-specific grammars. The multiple intermediate recognition results from the different recognizer-grammars are reconciled by running re-recognition using a dynamically composed grammar based on the multiple recognition results and potentially other domain knowledge, or selecting the winner using a statistical classifier operating on classification features extracted from the multiple recognition results and other domain knowledge.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
9 December 2010 (09.12.2010)(10) International Publication Number
WO 2010/141513 A3

(51) International Patent Classification:

G06F 17/00 (2006.01) *G06F 15/16* (2006.01)
G06F 17/18 (2006.01) *G06F 13/14* (2006.01)
G06F 17/27 (2006.01)

Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **BUNTSCHUH, Bruce**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(21) International Application Number:

PCT/US2010/036964

(22) International Filing Date:

1 June 2010 (01.06.2010)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

12/477,918 4 June 2009 (04.06.2009) US

(71) Applicant (for all designated States except US): **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

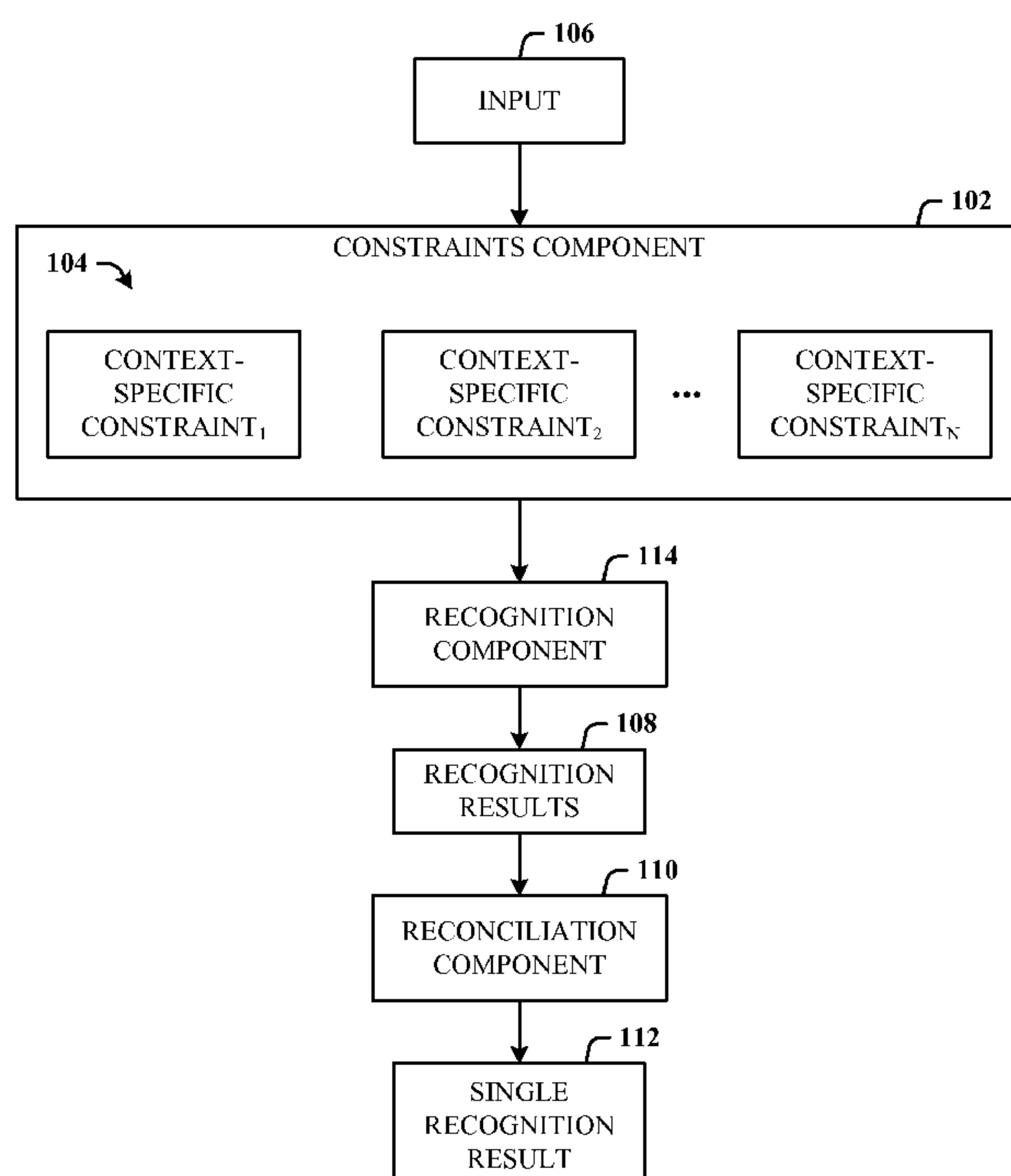
(72) Inventors: **CHANG, Shuangyu**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **LEVIT, Michael**; c/o Microsoft Corporation, LCA - International

(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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, 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, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK,

[Continued on next page]

(54) Title: RECOGNITION USING RE-RECOGNITION AND STATISTICAL CLASSIFICATION



(57) Abstract: Architecture that employs an overall grammar as a set of context-specific grammars for recognition of an input, each responsible for a specific context, such as subtask category, geographic region, etc. The grammars together cover the entire domain. Moreover, multiple recognitions can be run in parallel against the same input, where each recognition uses one or more of the context-specific grammars. The multiple intermediate recognition results from the different recognizer-grammars are reconciled by running re-recognition using a dynamically composed grammar based on the multiple recognition results and potentially other domain knowledge, or selecting the winner using a statistical classifier operating on classification features extracted from the multiple recognition results and other domain knowledge.

FIG. 1

WO 2010/141513 A3

SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(88) Date of publication of the international search report:

3 March 2011

RECOGNITION USING RE-RECOGNITION AND STATISTICAL CLASSIFICATION

BACKGROUND

[0001] Speech recognition performance is oftentimes suboptimal when a large grammar search space is involved, such as a voice search task that covers a large number of business names, web search queries, voice dialing requests, etc. Three main suboptimalities that are often exhibited include long recognition latency, poor recognition accuracy, and insufficient grammar coverage.

[0002] One existing mobile voice search application uses a nationwide business listing grammar plus a locality grammar at the first stage and re-recognizes the same utterance using a locality-specific business listing grammar at the second stage (where the locality was determined in the first stage). This approach does not address the latency issue, but can improve coverage and accuracy in very specific situations. Another approach attempts to reduce word error rate by voting among outputs of distinct recognizers at the sub-utterance level. The approach and its extensions generally assume each recognizer attempts recognition with a complete grammar for the entire task.

SUMMARY

[0003] The following presents a simplified summary in order to provide a basic understanding of some novel embodiments described herein. This summary is not an extensive overview, and it is not intended to identify key/critical elements or to delineate the scope thereof. Its sole purpose is to present some concepts in a simplified form as a prelude to the more detailed description that is presented later.

[0004] The disclosed architecture takes an input for recognition and applies different instances of context-specific constraints to the input for recognition processing. The separate constraint instances taken together provide the overall context domain for the given input. By operating recognition in parallel, for example, against these constraint instances, recognition latency, recognition accuracy, and recognition domain coverage are improved. Moreover, recognition processing of the separate recognition paths can be managed by imposing time limitations on how long the system will wait for a result to be produced.

[0005] In the context of speech recognition, the architecture employs an overall grammar in the form of a disjunction of smaller individual context-specific grammars for recognition of an utterance input, each responsible for a specific context, such as subtask

51331-1137

category, geographic region, etc. The grammars together cover the entire domain. Moreover, multiple recognitions can be run in parallel against the same input, where each recognition path uses one or more of the context-specific grammars.

5 [0006] The multiple intermediate recognition results from the different recognizer-grammars paths are reconciled by running re-recognition using a dynamically composed grammar based on the multiple recognition results and potentially other domain knowledge, or selecting the winner using a statistical classifier operating on classification features extracted from the multiple recognition results and other domain knowledge.

10 [0006a] According to an aspect of the present invention, there is provided a computer-implemented recognition system, comprising: a microprocessor; and a memory operatively coupled to the microprocessor, the memory having computer-executable instructions stored thereon for execution by the microprocessor, the computer-executable instructions comprising: code means for a constraints component of multiple context-specific constraints configured to perform independent recognition processing of a same input along multiple
15 recognition paths into respective multiple recognition results, wherein separate constraint instances taken together provide an overall context domain for the input; and code means for a reconciliation component configured to generate a dynamic grammar using the multiple recognition results, and configured to perform regression analysis to assign relative weights to each of the recognition paths in the dynamic grammar and to determine a final recognition
20 result.

[0006b] According to another aspect of the present invention, there is provided a computer-readable storage device having stored thereon computer-executable instructions for execution by a processor, the computer-executable instructions comprising: code means for a constraints component of multiple context-specific constraints configured to perform independent
25 recognition processing of a same input along multiple recognition paths into respective multiple recognition results, wherein separate constraint instances taken together provide an overall context domain for the input; and code means for a reconciliation component configured to generate a dynamic grammar using the multiple recognition results, and

51331-1137

configured to perform regression analysis to assign relative weights to each of the recognition paths in the dynamic grammar and to determine a final recognition result.

[0006c] According to still another aspect of the present invention, there is provided a computer-implemented recognition method, performed by a computer system executing machine-readable instructions, the method comprising acts of: receiving a recognition grammar composed of separate context-specific grammars each covering a specific subset of an original task space, for processing an utterance input; recognizing the utterance input in parallel paths using a corresponding context-specific grammar for each path; generating an intermediate recognition result from each path; generating a dynamic grammar utilizing the intermediate recognition result from each path; performing regression analysis to assign relative weights to each of the recognition paths in the dynamic grammar and to determine a final recognition result; and configuring a microprocessor to execute instructions in a memory associated with the acts of receiving, recognizing, generating the intermediate recognition, generating the dynamic grammar, and performing.

[0006d] According to yet another aspect of the present invention, there is provided a computer-implemented recognition system, comprising: a microprocessor; and a memory operatively coupled to the microprocessor, the memory having computer-executable instructions stored thereon for execution by the microprocessor, the computer-executable instructions comprising: code means for a constraints component of multiple context-specific constraints configured to perform independent recognition processing of an input along multiple recognition paths into respective multiple recognition results, wherein separate constraint instances taken together provide an overall context domain for the input; code means for a reconciliation component configured to generate a dynamic grammar using the multiple recognition results, and configured to perform statistical analysis to assign relative weights to each of the recognition paths in the dynamic grammar and to determine a final recognition result; and code means for a re-recognition component configured to process the input utilizing the dynamic grammar to generate the final recognition result.

[0006e] According to a further aspect of the present invention, there is provided a computer-implemented recognition method, performed by a computer system executing machine-

51331-1137

readable instructions, the method comprising acts of: recognizing an utterance input in parallel paths using a recognition grammar that comprises a different or intersecting context-specific grammar for each path; generating a dynamic grammar utilizing an intermediate recognition result from one or more of the paths; performing statistical analysis to assign relative weights

5 to each of the recognition paths in the dynamic grammar; performing re-recognition processing on the utterance input using the dynamic grammar to generate the final recognition result; and configuring a microprocessor to execute instructions in a memory associated with the acts of recognizing, generating, performing statistical analysis, and performing re-recognition processing.

51331-1137

[0007] To the accomplishment of the foregoing and related ends, certain illustrative aspects are described herein in connection with the following description and the annexed drawings. These aspects are indicative of the various ways in which the principles disclosed herein can be practiced and all aspects and equivalents thereof are intended to be within the scope of the claimed subject matter. Other advantages and novel features will become apparent from the following detailed description when considered in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 illustrates a computer-implemented recognition system in accordance with the disclosed architecture.

[0009] FIG. 2 illustrates an alternative embodiment of a system that employs rules for determination of the single recognition result.

[0010] FIG. 3 illustrates a context-specific constraints recognition system that employs re-recognition and where the constraints are grammars for voice recognition.

[0011] FIG. 4 illustrates a context-specific constraints recognition system that employs statistical classification and where the constraints are grammars for parallel voice recognition.

[0012] FIG. 5 illustrates a computer-implemented recognition method.

[0013] FIG. 6 illustrates further aspects of the method of FIG. 5.

[0014] FIG. 7 illustrates additional aspects of the method of FIG. 5.

[0015] FIG. 8 illustrates a block diagram of a computing system operable to execute recognition in accordance with the disclosed architecture.

[0016] FIG. 9 illustrates a schematic block diagram of a computing environment that provides parallel recognition in accordance with the disclosed architecture.

DETAILED DESCRIPTION

[0017] The disclosed architecture is a recognition system that first performs independent recognition of the same input (e.g., utterance) using context specific constraints. These independent recognitions can be performed either serially or in parallel. The context specific constraints are each smaller than a constraint that attempts to unify all domain knowledge. Reconciliation of the multiple recognition results can be accomplished using subsequent recognition (re-recognition) and/or via statistical classification.

[0018] The architecture addresses problems of recognition latency, recognition accuracy, and insufficient grammar coverage associated with a traditional single-grammar, single-recognition approach. With respect to recognition latency, each recognition instance in the parallel recognition is against a smaller grammar than a single large grammar that can cover the same tasks. Furthermore, the re-recognition step is against a small dynamic grammar. The two combined recognition stages of the maximum latency of the parallel recognitions, for example, plus the latency of re-recognition can have a smaller latency than recognition with single large grammar, particularly in non-streaming cases.

[0019] With a single recognition, recognition accuracy is oftentimes lost due to pruning during hypothesis search. Having multiple recognitions alleviates this limitation as a much larger hypothesis set can be maintained. In addition, context-specific constraints such as grammars are more likely to have better accuracy on utterances, for example, from the target context than a single, general grammar covering many contexts. Thus, there is a greater chance that the results from the multiple recognitions contain the correct result, and reconciling the multiple recognition results with re-recognition or a classifier is more likely to generate the correct result than a one-grammar, single-recognition approach.

[0020] With respect to insufficient grammar coverage, for example, there are oftentimes practical limitations (e.g., hardware, software) on how large a single grammar can be and/or how large grammars in a single recognition can be. Running multiple recognitions in parallel, for example, can significantly increase total grammar coverage, since each recognition can potentially run on distinct software processes and/or hardware resources.

[0021] Consider application of the disclosed architecture to a large-scale speech recognition task. The following example illustrates the concept by using a voice search task as an example, which can include an open-ended search of the web, local businesses, personal contacts, etc. Variants and enhancements are possible to various parts of the solution.

51331-1137

[0022] The recognition grammar is provided as a number of smaller and possibly overlapping context-specific grammars, each covering a specific subset of the original task space. The context for division can be based on subtask category (e.g., business names versus movie titles), geographic location (e.g., businesses in California versus in New York), demographic origins (e.g., youth oriented versus mature content), etc. Each content-specific grammar can be built separately, leveraging knowledge, structure and other available information relevant for each context to maximize the success rate for expected user inputs from each context.

[0023] Reference is now made to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding thereof. It may be evident, however, that the novel embodiments can be practiced without these specific details. In other instances, well known structures and devices are shown in block diagram form in order to facilitate a description thereof. The intention is to cover all modifications, equivalents, and alternatives falling within the scope of the claimed subject matter.

[0024] FIG. 1 illustrates a computer-implemented recognition system 100 in accordance with the disclosed architecture. The system 100 includes a constraints component 102 of context-specific constraints 104 for recognition processing of an input 106 into recognition results 108, and a reconciliation component 110 for reconciling the recognition results 108 into a single recognition result 112.

[0025] The system 100 can further comprise a recognition component 114 for separate recognition processing of corresponding context-specific constraints 104 in parallel paths and/or serially. For example, the context-specific constraints 104 can include grammars for recognition processing of the grammars against the input 106 in parallel paths and/or serial paths. The individual sets of context-specific constraints 104 can include disjointed and intersecting context coverage. In other words, one set of constraints can have some overlap with constraints of another constraint set. It is also the case where some constraints sets do not overlap with constraints of other constraint sets.

[0026] The reconciliation component 110 can reconcile the recognition results 108 using re-recognition to generate the single recognition result 112, by employing a dynamically composed grammar based on the recognition results 108.

[0027] Alternatively the reconciliation component 110 can reconcile the results 108 using a statistical classifier that operates on classification features extracted from the recognition results 108 to generate the single recognition result 112.

[0028] The reconciliation processing can also process relevant task data to arrive at the single recognition result 112. The relevant task data can include at least one of recognized strings, utterance level and sub-utterance level confidence scores, speech coverage, relative latencies among concurrent recognitions, prior probabilities of contexts, relative difficulty of each recognition, or consensus among the recognition results. In addition, a number of handcrafted and/or automatically derived rules reflecting specific requirements of the task can influence the reconciliation process of multiple recognition hypotheses.

[0029] FIG. 2 illustrates an alternative embodiment of a system 200 that employs rules for determination of the single recognition result 112. The system 200 includes the constraints component 102 of context-specific constraints 104 for recognition processing of the input 106 into the recognition results 108, and the reconciliation component 110 for reconciling the recognition results 108 into the single recognition result 112, and the recognition component 114 for separate recognition processing of corresponding context-specific constraints 104 in parallel paths and/or serially.

[0030] A rules component 202 is provided to apply rules (e.g., priority) for declaring one or more of the recognition results 108 and/or the single recognition result 112 (e.g., a final result). For example, a rule can be created and applied that determines if a specific recognizer returns a particular result with a sufficiently high confidence score, then that result can be accepted as final for that corresponding recognizer process or even for the single recognition result 112.

[0031] FIG. 3 illustrates a context-specific constraints recognition system 300 that employs re-recognition and where the constraints are grammars for voice recognition. The system 300 includes N recognition-grammar pairs operating in parallel, where each pair includes one or more context-specific grammars and a recognizer (denoted Recognition N). As illustrated, the grammars are different; however, there may be some overlap of one grammar to another grammar, although this is not necessary. Rather than creating and utilizing one large grammar, as in existing recognition systems, the system 300 retains the separate grammars (instead of merging into one large grammar), and runs recognition of a speech utterance input 302 on each of the grammars.

[0032] In other words, the utterance input 302 is processed through a first recognizer 304 and associated first context-specific grammar 306 producing first result(s) 308, as well as

through a second recognizer 310 and associated second context-specific grammar 312 producing second result(s) 314, and so on, to the desired number N of recognizers and grammars thereby producing N result(s). The result(s) are utilized to generate a dynamic grammar 316, which can then be used for re-recognition 318 to output a final recognition result 320.

[0033] Put another way, with the user utterance input 302, a separate recognition is run against each of the context-specific grammars. This is illustrated as occurring in a parallel fashion at the same time or approximately the same time. Each of the parallel recognitions can employ the same kind or a different kind of recognizer (e.g., embedded versus network recognizers, network recognizers with different acoustic models, etc.), and use the same or different recognition parameters. Up to a maximum waiting period, the system 300 collects all available recognition results (e.g., result(s) 308, result(s) 314, etc.) and determines the final recognition result 320 by re-recognition.

[0034] The dynamic grammar 316 is constructed to include competing entries derived from all recognition results, which can include recognition strings, interpretations, and confidence scores, of the N-best recognition results, and/or recognition lattice, if available. Re-recognition of the original utterance input 302 is performed against this dynamic grammar 316. The result of the re-recognition 318, including confidence scores, is taken as the final recognition result 320.

[0035] Optionally, certain priority rules can be included by the rules component 202 to declare the final recognition result 320 before all recognitions are completed, such as if a certain recognizer returns a particular result with a sufficiently high confidence score, this result can be accepted as final. Optionally, other domain knowledge 322 that is relevant to the task can be provided as input to the dynamic grammar to provide a more focused recognition process. This knowledge 322 can include user preferences, content related to what is being said in the utterance, hardware/software considerations, locality, and so on.

[0036] FIG. 4 illustrates a context-specific constraints recognition system 400 that employs statistical classification and where the constraints are grammars for voice recognition in parallel. Multiple numerical and/or categorical features 402 can be derived from all recognition results (e.g., result(s) 308, result(s) 314, etc.), and potentially, the other domain knowledge 322 relevant for the recognition task. A statistical classifier is used to determine how likely each result reflects the actual user input. The result with the highest classification score can be selected as the final recognition result 320 and the classification score can be normalized to be the final recognition confidence.

[0037] The system 400 includes the N recognition-grammar pairs operating in parallel, where each pair includes a context-specific grammar (denoted Context-Specific Grammar N) and a recognizer (denoted Recognition N). As previously illustrated and described, the grammars are different; however, there may be some overlap of one grammar to another grammar, although this is not necessary. Rather than creating and utilizing one large grammar, as in existing recognition systems, the system 400 retains the separate grammars (instead of merging into one large grammar), and runs recognition of the speech utterance input 302 on each of the grammars.

[0038] In other words, the utterance input 302 is processed through the first recognizer 304 and associated first context-specific grammar 306 producing the first result(s) 308, as well as through a second recognizer 310 and associated second context-specific grammar 312 producing the second result(s) 314, and so on, to the desired number N of recognizers and grammars thereby producing N result(s). The result(s) (Result(s) 308, Result(s) 314,...,Result(s) N) are utilized to generate features 402, which are then passed to statistical classification 404 for the final recognition result 320.

[0039] As previously illustrated and described in FIG. 3, optionally, certain priority rules can be included by the rules component 202 to declare the final recognition result 320 before all recognitions are completed, such as if a certain recognizer returns a particular result with a sufficiently high confidence score, this result can be accepted as final. Optionally, the other domain knowledge 322 that is relevant to the task can be provided as input to the dynamic grammar to provide a more focused recognition process. This knowledge 322 can include user preferences, content related to what is being said in the utterance, hardware/software considerations, locality, and so on.

[0040] Note that the description herein covers how the architecture works when receiving a user input utterance online. Another aspect of the solution is to choose the appropriate settings, features, etc., used by the system, particularly during the reconciliation of multiple recognition results. For both the re-recognition and the statistical classifier approaches, training data can be utilized and an offline training process can be employed to select an optimal configuration and parameterization.

[0041] For the re-recognition approach, it is also possible to optionally perform a statistical analysis such as regression to assign relative weights to paths in the re-recognition dynamic grammar. The output of the other domain knowledge 322 can be controlled to influence the dynamic grammar 316 for each re-recognition process.

[0042] In either approach, one or more of the following features 402 can be employed, some features directly obtained from parallel recognition results, and other features derived from relevant task knowledge. The features 402 can include, but are not limited to, recognized strings, utterance level and sub-utterance level confidence scores, speech coverage (e.g., fraction of an utterance hypothesized as speech), relative latencies among the recognitions (e.g., parallel), prior probabilities of contexts (e.g., how often users ask for business names versus sports scores), relative difficulty of each context-specific recognition (e.g., perplexity of the context-specific grammars, within context recognition accuracy), admissibility of each grammar (e.g., web search grammar can accept a large variety of queries), and consensus among the recognition results.

[0043] Note that the individual recognition processes can be distributed across different machines such as server, clients, or a combination of servers and clients. This applies to parallel recognition as well as serial recognition in both classification and re-recognition scenarios.

[0044] Put another way, the disclosed architecture is a computer-implemented recognition system that comprises the constraints component of context-specific grammars for recognition processing of an utterance input into recognition results, the recognition component for individual recognition processing of the utterance input in parallel paths using corresponding context-specific grammars, and the reconciliation component for reconciling the recognition results into a final recognition result.

[0045] The reconciliation component employs a dynamically composed grammar of the recognition results and reconciles the recognition results using re-recognition to generate the final recognition result. Optionally, the reconciliation component reconciles the recognition results by employing statistical analysis such as regression prior to re-recognition to determine the final recognition result. Alternatively, the reconciliation component reconciles the recognition results using statistical classification that operates on features extracted from the recognition results to generate the final recognition result. In addition, the rules component imposes one or more rules that define determination of the final recognition result and, other domain knowledge can influence features for statistical classification reconciliation and a dynamic grammar for re-recognition reconciliation.

[0046] Included herein is a set of flow charts representative of exemplary methodologies for performing novel aspects of the disclosed architecture. While, for purposes of simplicity of explanation, the one or more methodologies shown herein, for example, in

the form of a flow chart or flow diagram, are shown and described as a series of acts, it is to be understood and appreciated that the methodologies are not limited by the order of acts, as some acts may, in accordance therewith, occur in a different order and/or concurrently with other acts from that shown and described herein. For example, those skilled in the art will understand and appreciate that a methodology could alternatively be represented as a series of interrelated states or events, such as in a state diagram. Moreover, not all acts illustrated in a methodology may be required for a novel implementation.

[0047] FIG. 5 illustrates a computer-implemented recognition method. At 500, separate context-specific grammars are received for processing an utterance input. At 502, the utterance input is recognized in parallel paths using a corresponding context-specific grammar for each path. At 504, an intermediate recognition result is generated from each path. At 506, the intermediate recognition results are reconciled into a final recognition result.

[0048] FIG. 6 illustrates further aspects of the method of FIG. 5. At 600, the intermediate recognition results are reconciled using re-recognition of a dynamic grammar generated from the recognition results. At 602, other domain knowledge is input during reconciliation of the intermediate recognition results by re-recognition. At 604, one or more rules are imposed to generate the final recognition result. At 606, other domain knowledge is input during reconciliation of the intermediate recognition results by statistical classification. At 608, one or more rules are imposed to generate the final recognition result.

[0049] FIG. 7 illustrates additional aspects of the method of FIG. 5. At 700, statistical analysis such as regression is performed. The analysis is carried out over all paths concurrently. At 702, relative weights are assigned to each path in a re-recognition dynamic grammar. At 704, a predetermined amount of time is waited for generation of an intermediate recognition result of a path. At 706, the final recognition result is generated based on intermediate recognition results that are generated within the amount of time.

[0050] As used in this application, the terms “component” and “system” are intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component can be, but is not limited to being, a process running on a processor, a processor, a hard disk drive, multiple storage drives (of optical, solid state, and/or magnetic storage medium), an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration,

both an application running on a server and the server can be a component. One or more components can reside within a process and/or thread of execution, and a component can be localized on one computer and/or distributed between two or more computers. The word “exemplary” may be used herein to mean serving as an example, instance, or illustration. Any aspect or design described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects or designs.

[0051] Referring now to FIG. 8, there is illustrated a block diagram of a computing system 800 operable to execute recognition in accordance with the disclosed architecture. In order to provide additional context for various aspects thereof, FIG. 8 and the following discussion are intended to provide a brief, general description of the suitable computing system 800 in which the various aspects can be implemented. While the description above is in the general context of computer-executable instructions that can run on one or more computers, those skilled in the art will recognize that a novel embodiment also can be implemented in combination with other program modules and/or as a combination of hardware and software.

[0052] The computing system 800 for implementing various aspects includes the computer 802 having processing unit(s) 804, a system memory 806, and a system bus 808. The processing unit(s) 804 can be any of various commercially available processors such as single-processor, multi-processor, single-core units and multi-core units. Moreover, those skilled in the art will appreciate that the novel methods can be practiced with other computer system configurations, including minicomputers, mainframe computers, as well as personal computers (e.g., desktop, laptop, etc.), hand-held computing devices, microprocessor-based or programmable consumer electronics, and the like, each of which can be operatively coupled to one or more associated devices.

[0053] The system memory 806 can include volatile (VOL) memory 810 (e.g., random access memory (RAM)) and non-volatile memory (NON-VOL) 812 (e.g., ROM, EPROM, EEPROM, etc.). A basic input/output system (BIOS) can be stored in the non-volatile memory 812, and includes the basic routines that facilitate the communication of data and signals between components within the computer 802, such as during startup. The volatile memory 810 can also include a high-speed RAM such as static RAM for caching data.

[0054] The system bus 808 provides an interface for system components including, but not limited to, the memory subsystem 806 to the processing unit(s) 804. The system bus 808 can be any of several types of bus structure that can further interconnect to a memory

bus (with or without a memory controller), and a peripheral bus (e.g., PCI, PCIe, AGP, LPC, etc.), using any of a variety of commercially available bus architectures.

[0055] The computer 802 further includes storage subsystem(s) 814 and storage interface(s) 816 for interfacing the storage subsystem(s) 814 to the system bus 808 and other desired computer components. The storage subsystem(s) 814 can include one or more of a hard disk drive (HDD), a magnetic floppy disk drive (FDD), and/or optical disk storage drive (e.g., a CD-ROM drive DVD drive), for example. The storage interface(s) 816 can include interface technologies such as EIDE, ATA, SATA, and IEEE 1394, for example.

[0056] One or more programs and data can be stored in the memory subsystem 806, a removable memory subsystem 818 (e.g., flash drive form factor technology), and/or the storage subsystem(s) 814 (e.g., optical, magnetic, solid state), including an operating system 820, one or more application programs 822, other program modules 824, and program data 826.

[0057] The one or more application programs 822, other program modules 824, and program data 826 can include the components, entities, and results of the system 100 of FIG. 1, the components, entities, and results of the system 200 of FIG. 2, the components, entities, and results of the system 300 of FIG. 3, the components, entities, and results of the system 400 of FIG. 4, and the methods and additional aspects provided in Figures 5-7, for example.

[0058] Generally, programs include routines, methods, data structures, other software components, etc., that perform particular tasks or implement particular abstract data types. All or portions of the operating system 820, applications 822, modules 824, and/or data 826 can also be cached in memory such as the volatile memory 810, for example. It is to be appreciated that the disclosed architecture can be implemented with various commercially available operating systems or combinations of operating systems (e.g., as virtual machines).

[0059] The storage subsystem(s) 814 and memory subsystems (806 and 818) serve as computer readable media for volatile and non-volatile storage of data, data structures, computer-executable instructions, and so forth. Computer readable media can be any available media that can be accessed by the computer 802 and includes volatile and non-volatile media, removable and non-removable media. For the computer 802, the media accommodate the storage of data in any suitable digital format. It should be appreciated by those skilled in the art that other types of computer readable media can be employed

such as zip drives, magnetic tape, flash memory cards, cartridges, and the like, for storing computer executable instructions for performing the novel methods of the disclosed architecture.

[0060] A user can interact with the computer 802, programs, and data using external user input devices 828 such as a keyboard and a mouse. Other external user input devices 828 can include a microphone, an IR (infrared) remote control, a joystick, a game pad, camera recognition systems, a stylus pen, touch screen, gesture systems (e.g., eye movement, head movement, etc.), and/or the like. The user can interact with the computer 802, programs, and data using onboard user input devices 830 such as a touchpad, microphone, keyboard, etc., where the computer 802 is a portable computer, for example. These and other input devices are connected to the processing unit(s) 804 through input/output (I/O) device interface(s) 832 via the system bus 808, but can be connected by other interfaces such as a parallel port, IEEE 1394 serial port, a game port, a USB port, an IR interface, etc. The I/O device interface(s) 832 also facilitate the use of output peripherals 834 such as printers, audio devices, camera devices, and so on, such as a sound card and/or onboard audio processing capability.

[0061] One or more graphics interface(s) 836 (also commonly referred to as a graphics processing unit (GPU)) provide graphics and video signals between the computer 802 and external display(s) 838 (e.g., LCD, plasma) and/or onboard displays 840 (e.g., for portable computer). The graphics interface(s) 836 can also be manufactured as part of the computer system board.

[0062] The computer 802 can operate in a networked environment (e.g., IP) using logical connections via a wired/wireless communications subsystem 842 to one or more networks and/or other computers. The other computers can include workstations, servers, routers, personal computers, microprocessor-based entertainment appliance, a peer device or other common network node, and typically include many or all of the elements described relative to the computer 802. The logical connections can include wired/wireless connectivity to a local area network (LAN), a wide area network (WAN), hotspot, and so on. LAN and WAN networking environments are commonplace in offices and companies and facilitate enterprise-wide computer networks, such as intranets, all of which may connect to a global communications network such as the Internet.

[0063] When used in a networking environment the computer 802 connects to the network via a wired/wireless communication subsystem 842 (e.g., a network interface adapter, onboard transceiver subsystem, etc.) to communicate with wired/wireless networks,

wired/wireless printers, wired/wireless input devices 844, and so on. The computer 802 can include a modem or has other means for establishing communications over the network. In a networked environment, programs and data relative to the computer 802 can be stored in the remote memory/storage device, as is associated with a distributed system. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers can be used.

[0064] The computer 802 is operable to communicate with wired/wireless devices or entities using the radio technologies such as the IEEE 802.xx family of standards, such as wireless devices operatively disposed in wireless communication (e.g., IEEE 802.11 over-the-air modulation techniques) with, for example, a printer, scanner, desktop and/or portable computer, personal digital assistant (PDA), communications satellite, any piece of equipment or location associated with a wirelessly detectable tag (e.g., a kiosk, news stand, restroom), and telephone. This includes at least Wi-Fi (or Wireless Fidelity) for hotspots, WiMax, and Bluetooth™ wireless technologies. Thus, the communications can be a predefined structure as with a conventional network or simply an ad hoc communication between at least two devices. Wi-Fi networks use radio technologies called IEEE 802.11x (a, b, g, etc.) to provide secure, reliable, fast wireless connectivity. A Wi-Fi network can be used to connect computers to each other, to the Internet, and to wire networks (which use IEEE 802.3-related media and functions).

[0065] Referring now to FIG. 9, there is illustrated a schematic block diagram of a computing environment 900 that provides parallel recognition in accordance with the disclosed architecture. The environment 900 includes one or more client(s) 902. The client(s) 902 can be hardware and/or software (e.g., threads, processes, computing devices). The client(s) 902 can house cookie(s) and/or associated contextual information, for example.

[0066] The environment 900 also includes one or more server(s) 904. The server(s) 904 can also be hardware and/or software (e.g., threads, processes, computing devices). The servers 904 can house threads to perform transformations by employing the architecture, for example. One possible communication between a client 902 and a server 904 can be in the form of a data packet adapted to be transmitted between two or more computer processes. The data packet may include a cookie and/or associated contextual information, for example. The environment 900 includes a communication framework 906 (e.g., a global communication network such as the Internet) that can be employed to facilitate communications between the client(s) 902 and the server(s) 904.

51331-1137

[0067] Communications can be facilitated via a wire (including optical fiber) and/or wireless technology. The client(s) 902 are operatively connected to one or more client data store(s) 908 that can be employed to store information local to the client(s) 902 (e.g., cookie(s) and/or associated contextual information). Similarly, the server(s) 904 are operatively connected to one or more server data store(s) 910 that can be employed to store information local to the servers 904.

[0068] The client(s) 902 can include a client via which voice signals are received for recognition processing by the server(s) 904 or other client(s) 902. The grammars can be stored in the client datastore(s) 908 and/or the server datastore(s) 910.

[0069] What has been described above includes examples of the disclosed architecture. It is, of course, not possible to describe every conceivable combination of components and/or methodologies, but one of ordinary skill in the art may recognize that many further combinations and permutations are possible. Accordingly, the novel architecture is intended to embrace all such alterations, modifications and variations that fall within the scope of the appended claims. Furthermore, to the extent that the term “includes” is used in either the detailed description or the claims, such term is intended to be inclusive in a manner similar to the term “comprising” as “comprising” is interpreted when employed as a transitional word in a claim.

51331-1137

CLAIMS:

1. A computer-implemented recognition system, comprising:

a microprocessor; and

a memory operatively coupled to the microprocessor, the memory having
5 computer-executable instructions stored thereon for execution by the microprocessor, the
computer-executable instructions comprising:

code means for a constraints component of multiple context-specific
constraints configured to perform independent recognition processing of a same input along
multiple recognition paths into respective multiple recognition results, wherein separate
10 constraint instances taken together provide an overall context domain for the input; and

code means for a reconciliation component configured to generate a dynamic
grammar using the multiple recognition results, and configured to perform regression analysis
to assign relative weights to each of the recognition paths in the dynamic grammar and to
determine a final recognition result.
- 15 2. The system of claim 1, wherein the constraints include grammars for
recognition processing of the input in parallel paths.
3. The system of claim 1, wherein the reconciliation component reconciles the
results using re-recognition to generate the final recognition result.
4. The system of claim 3, wherein the re-recognition employs the dynamically
20 composed grammar based on the recognition results.
5. The system of claim 1, wherein the reconciliation component reconciles the
results using a statistical classifier that operates on classification features extracted from the
recognition results to generate the final recognition result.
6. The system of claim 1, wherein the context-specific constraints include
25 disjointed and intersecting context coverage.

51331-1137

7. The system of claim 1, wherein the recognition processing processes relevant task data to arrive at the final recognition result, the relevant task data includes at least one of recognized strings, utterance level and sub-utterance level confidence scores, speech coverage, relative latencies among concurrent recognitions, prior probabilities of contexts,
 5 relative difficulty of each recognition, or consensus among the recognition results.

8. The system of claim 1, further comprising a recognition component configured for separate recognition processing of the input using a corresponding context-specific constraint in each of parallel paths.

9. The system of claim 1, further comprising a rules component configured to
 10 impose one or more rules that define determination of the final recognition result.

10. A computer-readable storage device having stored thereon computer-executable instructions for execution by a processor, the computer-executable instructions comprising:

code means for a constraints component of multiple context-specific
 15 constraints configured to perform independent recognition processing of a same input along multiple recognition paths into respective multiple recognition results, wherein separate constraint instances taken together provide an overall context domain for the input; and

code means for a reconciliation component configured to generate a dynamic grammar using the multiple recognition results, and configured to perform regression analysis
 20 to assign relative weights to each of the recognition paths in the dynamic grammar and to determine a final recognition result.

11. The computer-readable storage device of claim 10, wherein the reconciliation component employs the dynamically composed grammar of the recognition results and reconciles the recognition results using re-recognition to generate the final recognition result.

25 12. The computer-readable storage device of claim 10, wherein the reconciliation component reconciles the recognition results using classification that operates on features extracted from the recognition results to generate the final recognition result.

51331-1137

13. The computer-readable storage device of claim 10, further comprising code means for a rules component for imposing one or more rules that define determination of the final recognition result and, other domain knowledge that influences features for classification reconciliation and a dynamic grammar for re-recognition reconciliation.

5 14. The computer-readable storage device of claim 10, wherein the reconciliation component reconciles the recognition results by employing regression analysis prior to re-recognition to determine the final recognition result.

15. A computer-implemented recognition method, performed by a computer system executing machine-readable instructions, the method comprising acts of:

10 receiving a recognition grammar composed of separate context-specific grammars each covering a specific subset of an original task space, for processing an utterance input;

recognizing the utterance input in parallel paths using a corresponding context-specific grammar for each path;

15 generating an intermediate recognition result from each path;

generating a dynamic grammar utilizing the intermediate recognition result from each path;

performing regression analysis to assign relative weights to each of the recognition paths in the dynamic grammar and to determine a final recognition result; and

20 configuring a microprocessor to execute instructions in a memory associated with the acts of receiving, recognizing, generating the intermediate recognition, generating the dynamic grammar, and performing.

16. The method of claim 15, further comprising reconciling the intermediate recognition results using re-recognition of the dynamic grammar generated from the
25 intermediate recognition results.

51331-1137

17. The method of claim 15, further comprising:

inputting other domain knowledge during reconciliation of the intermediate recognition results by re-recognition; and

imposing one or more rules to generate the final recognition result.

5 18. The method of claim 15, further comprising:

inputting other domain knowledge during reconciliation of the intermediate recognition results by classification; and

imposing one or more rules to generate the final recognition result.

19. The method of claim 15, further comprising:

10 performing classification analysis; and

assigning relative weights to each path in a re-recognition dynamic grammar.

20. The method of claim 15, further comprising:

waiting for a predetermined amount of time for generation of an intermediate recognition result of a path; and

15 generating the final recognition result based on intermediate recognition results that are generated within the amount of time.

21. A computer-implemented recognition system, comprising:

a microprocessor; and

20 a memory operatively coupled to the microprocessor, the memory having computer-executable instructions stored thereon for execution by the microprocessor, the computer-executable instructions comprising:

51331-1137

code means for a constraints component of multiple context-specific constraints configured to perform independent recognition processing of an input along multiple recognition paths into respective multiple recognition results, wherein separate constraint instances taken together provide an overall context domain for the input;

- 5 code means for a reconciliation component configured to generate a dynamic grammar using the multiple recognition results, and configured to perform statistical analysis to assign relative weights to each of the recognition paths in the dynamic grammar and to determine a final recognition result; and

- code means for a re-recognition component configured to process the input
10 utilizing the dynamic grammar to generate the final recognition result.

22. A computer-implemented recognition method, performed by a computer system executing machine-readable instructions, the method comprising acts of:

recognizing an utterance input in parallel paths using a recognition grammar that comprises a different or intersecting context-specific grammar for each path;

- 15 generating a dynamic grammar utilizing an intermediate recognition result from one or more of the paths;

performing statistical analysis to assign relative weights to each of the recognition paths in the dynamic grammar;

- performing re-recognition processing on the utterance input using the dynamic
20 grammar to generate the final recognition result; and

configuring a microprocessor to execute instructions in a memory associated with the acts of recognizing, generating, performing statistical analysis, and performing re-recognition processing.

100 ↙

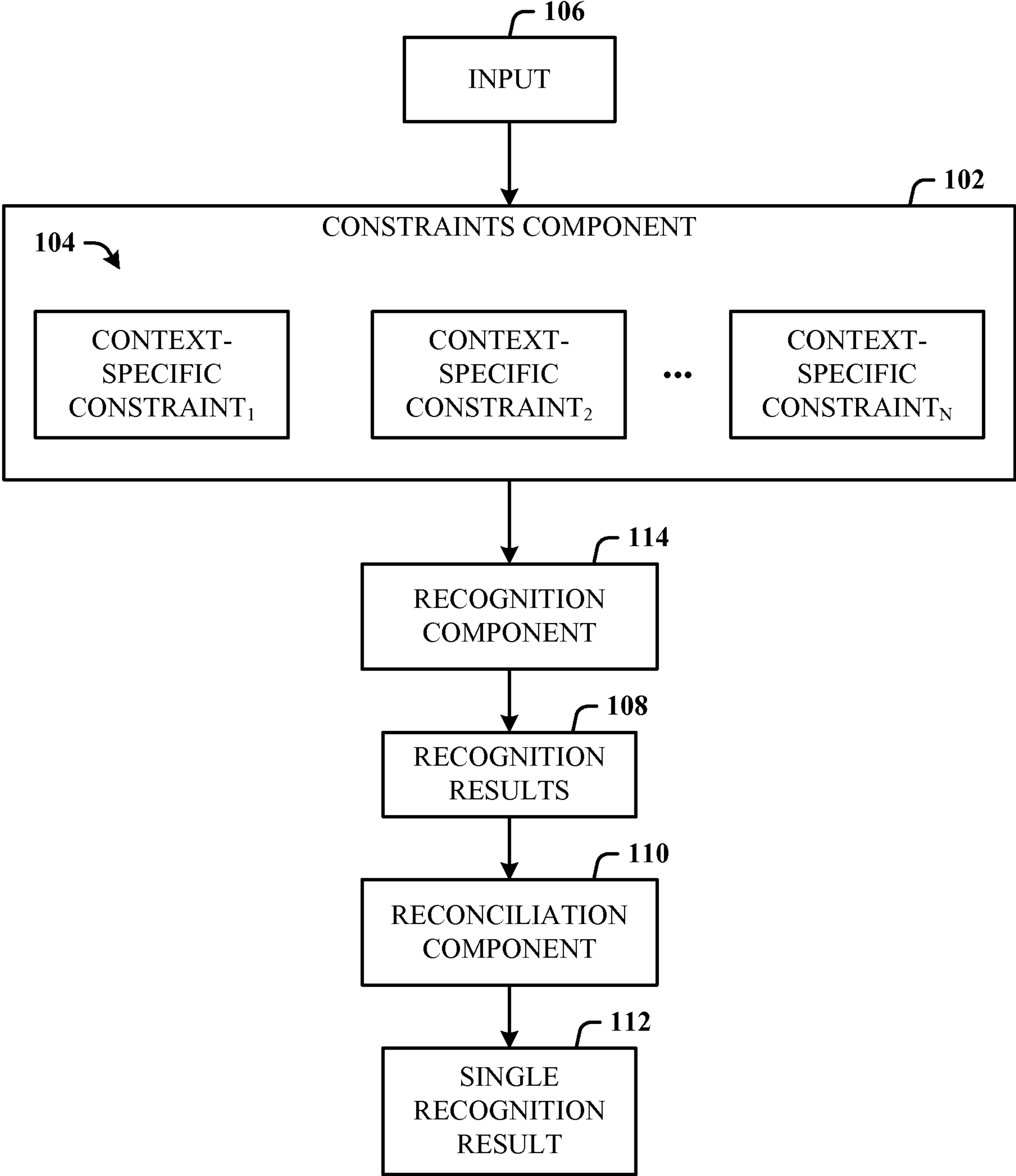


FIG. 1

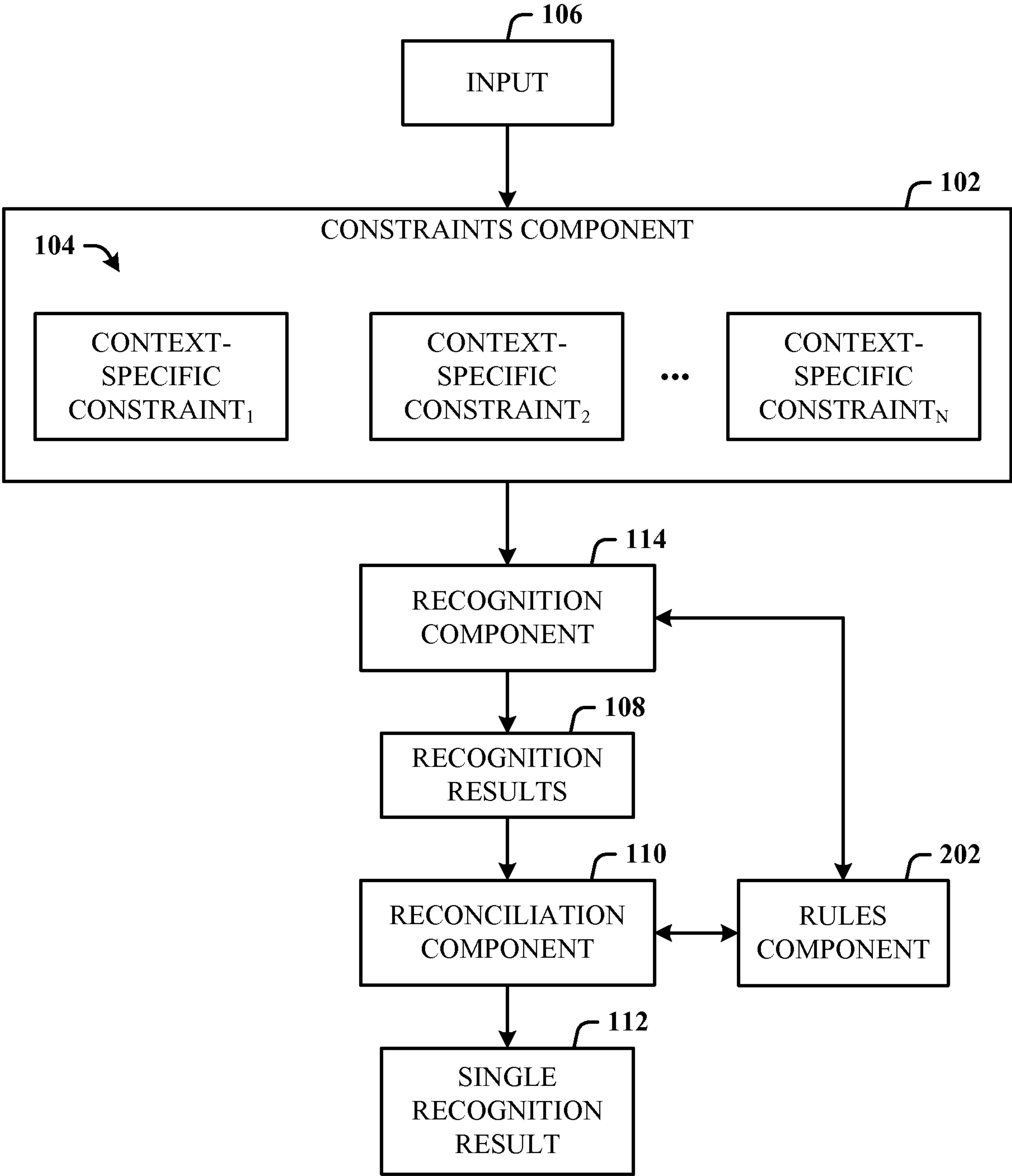


FIG. 2

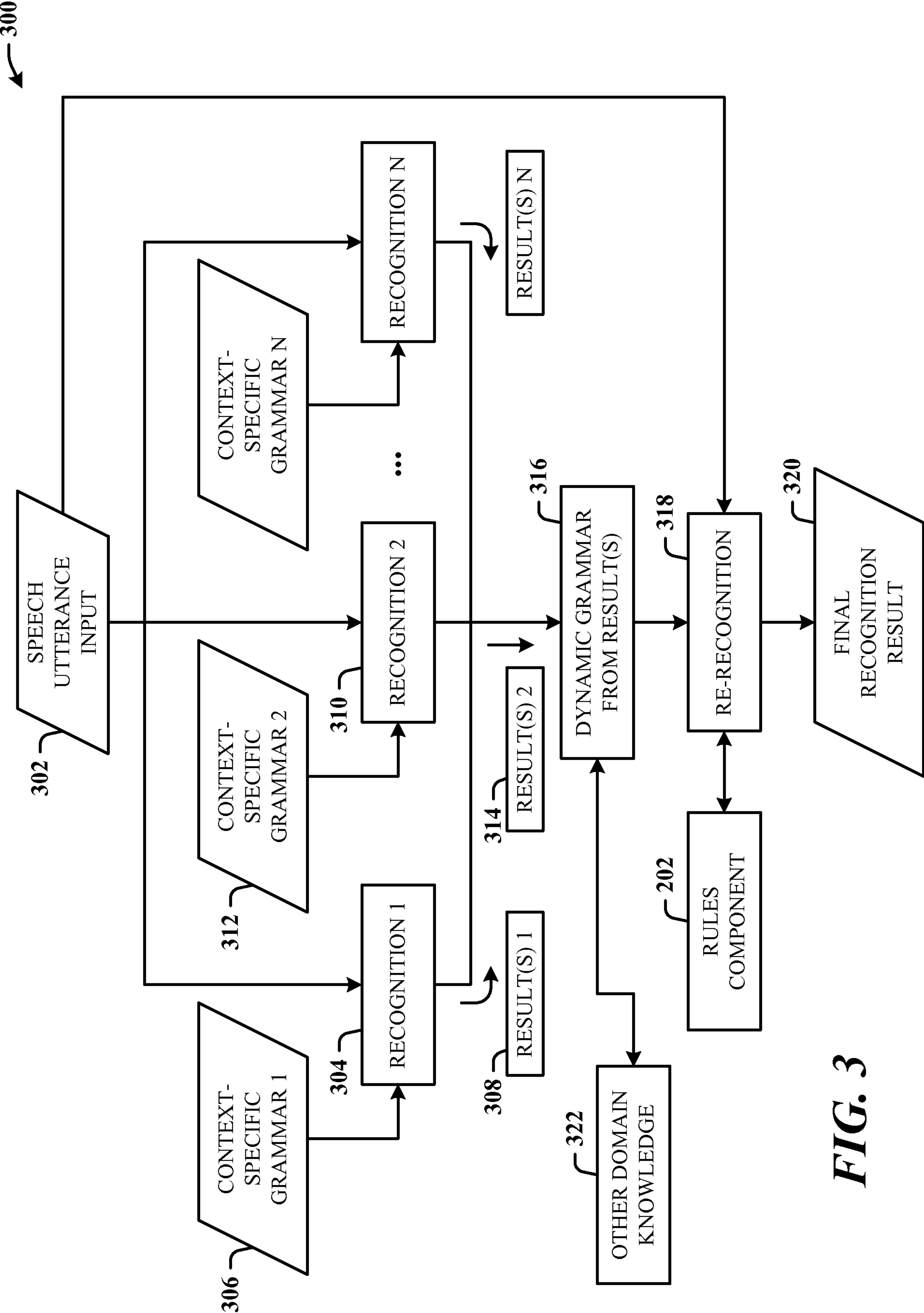
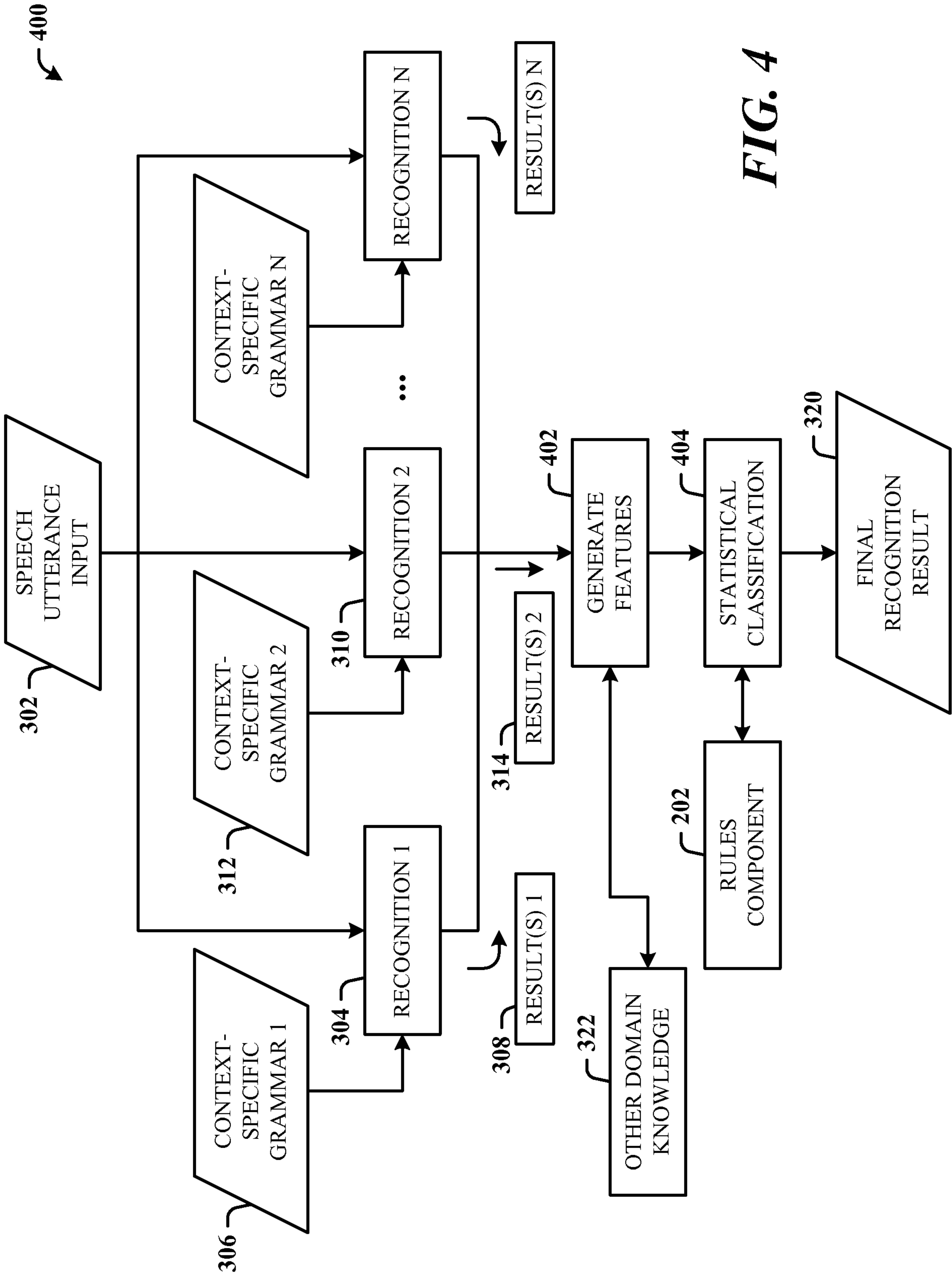
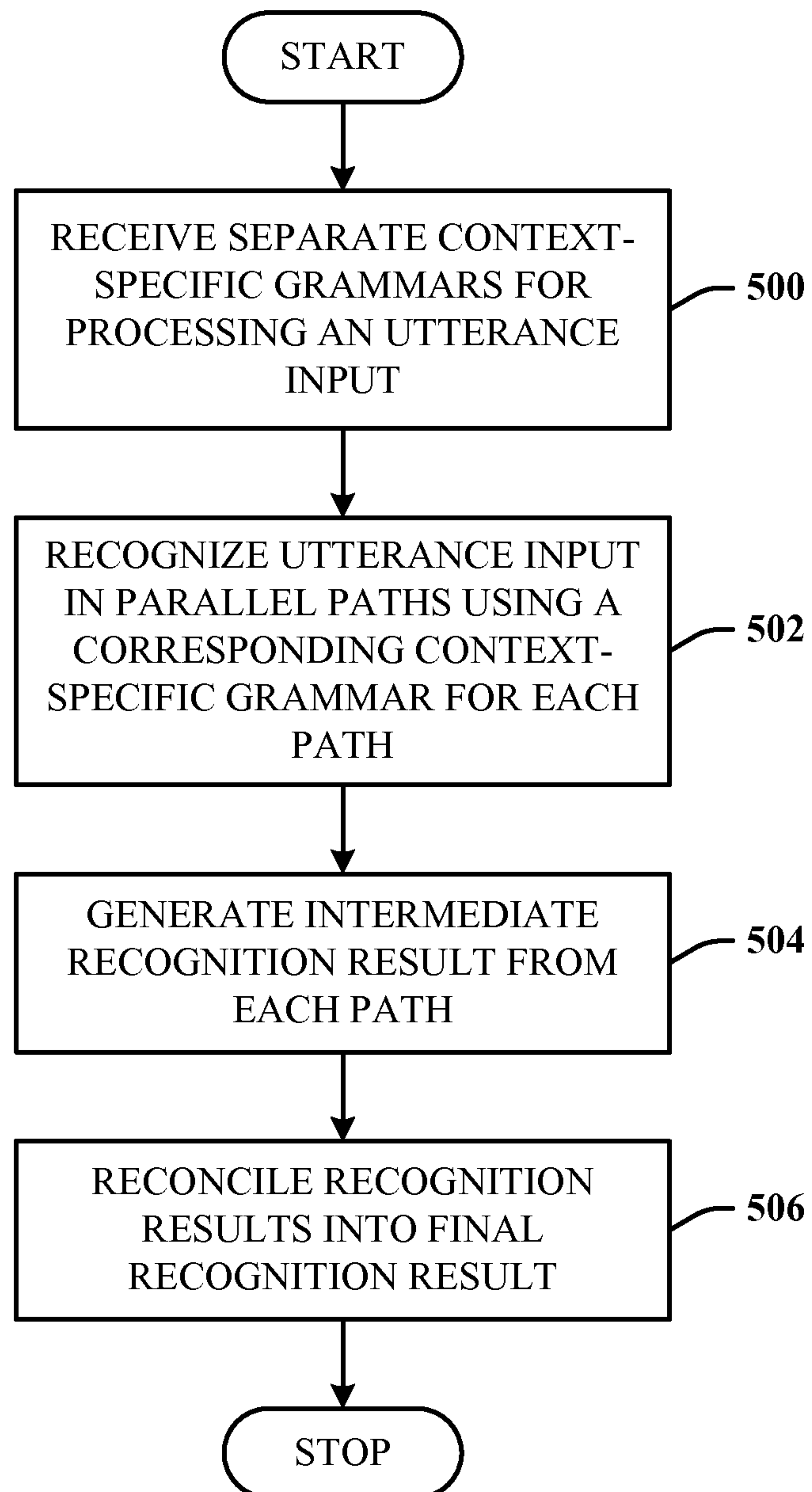


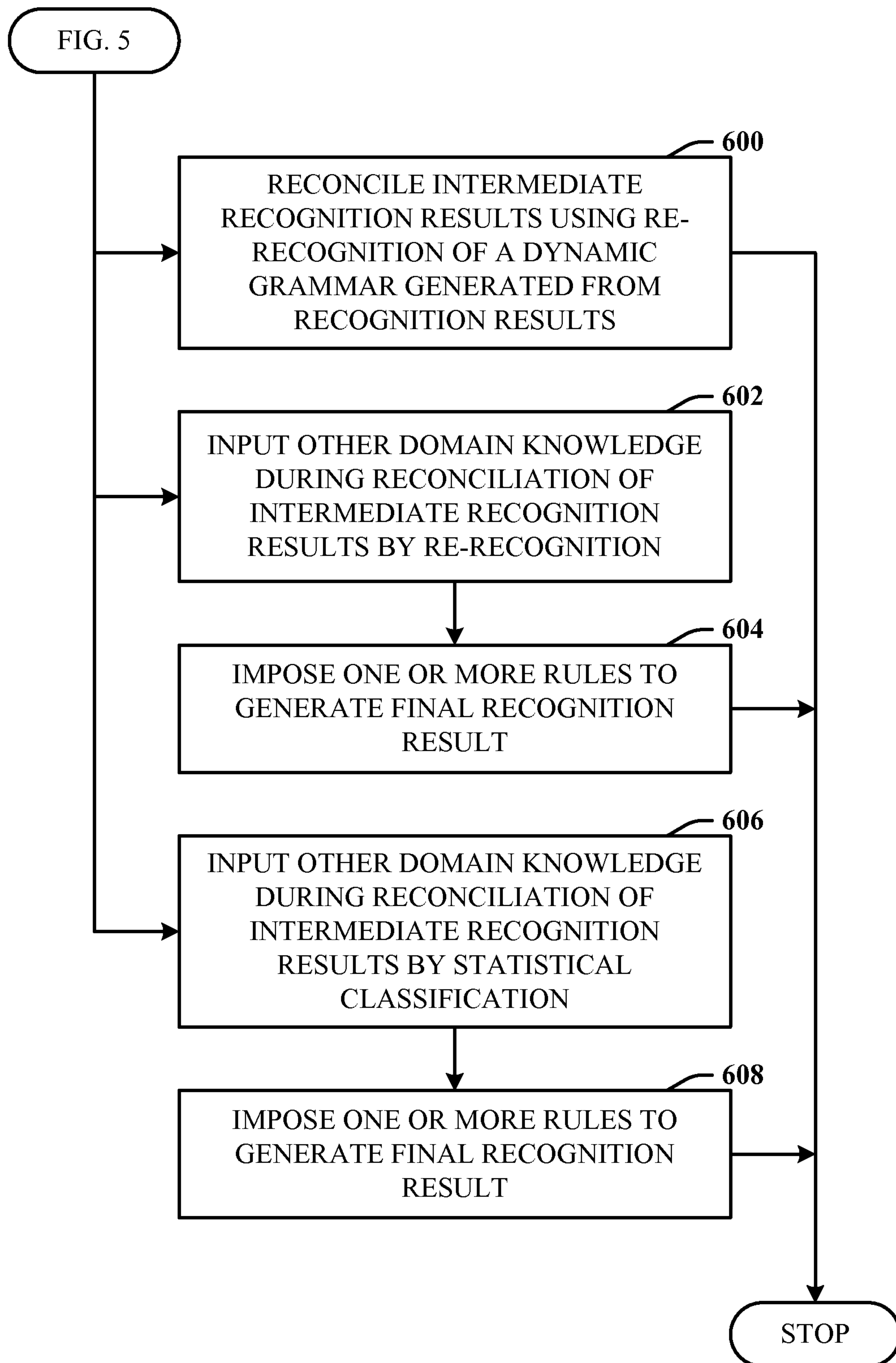
FIG. 3



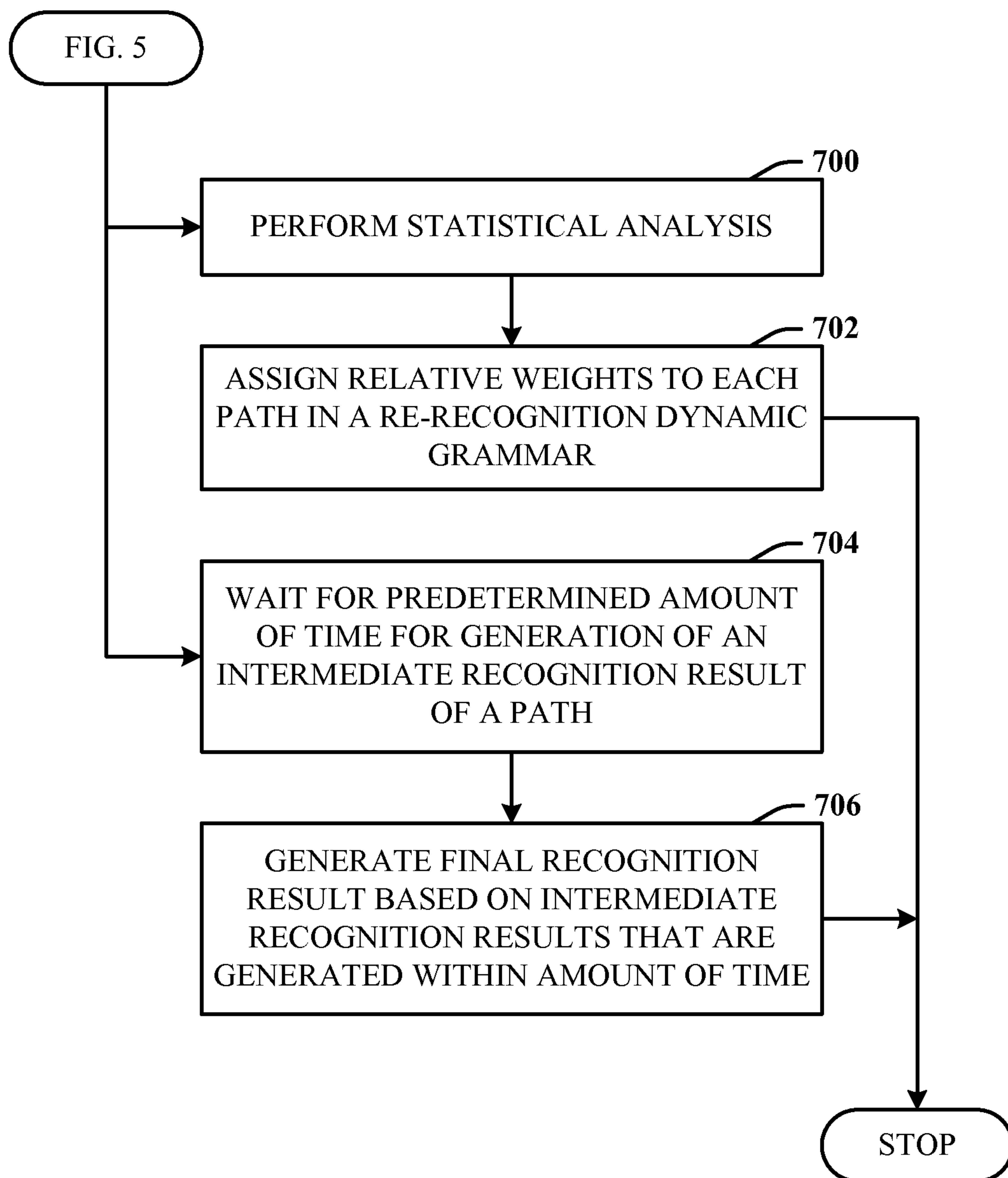
5/9

**FIG. 5**

6/9

**FIG. 6**

7/9

**FIG. 7**

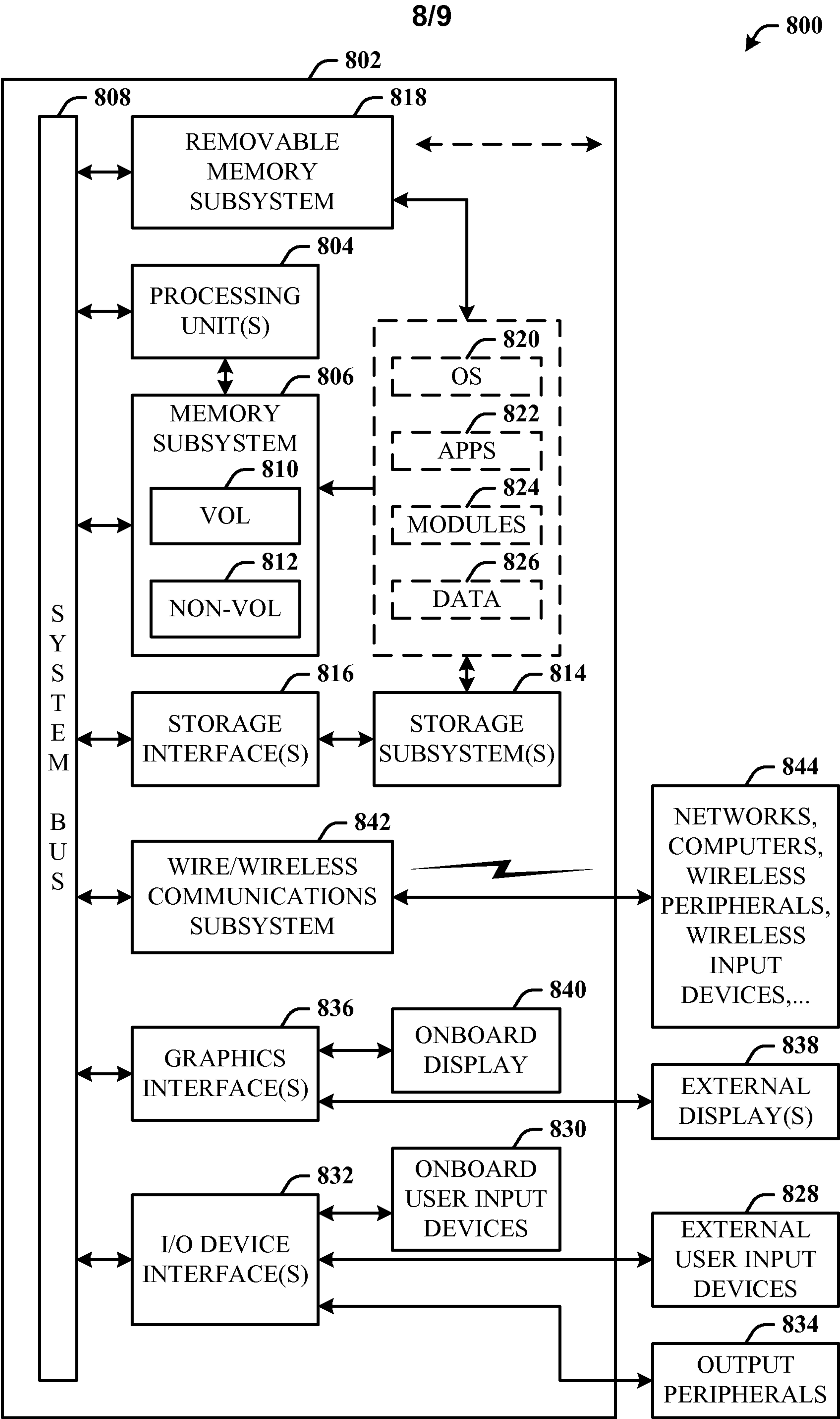


FIG. 8

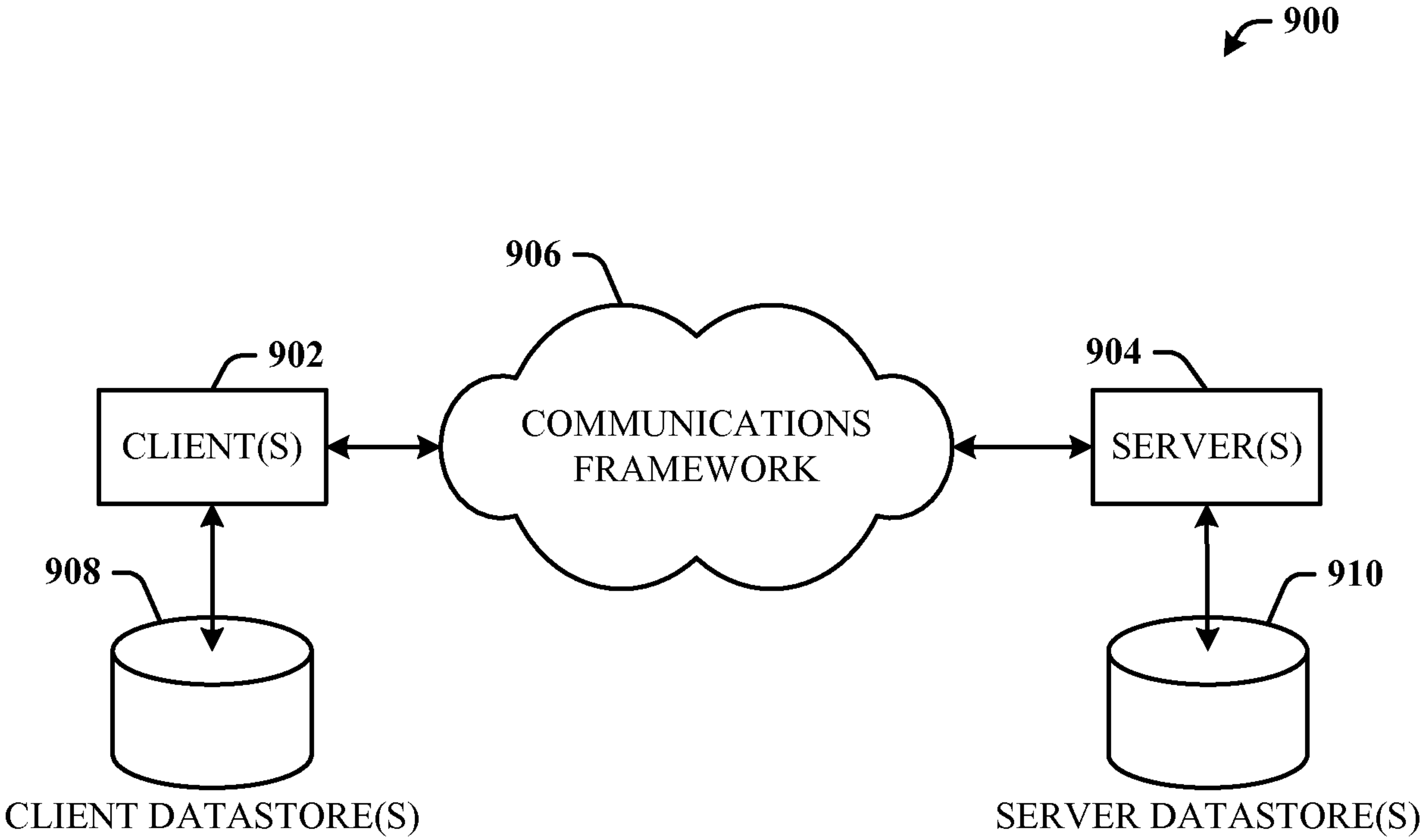


FIG. 9