

Filed: August 15, 2025

Filed on behalf of Meta Platforms, Inc.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

META PLATFORMS, INC.,
Petitioner,

v.

DIALECT, LLC,
Patent Owner

Case No. IPR2025-01336
U.S. Patent No. 9,734,825

**DECLARATION OF DANIEL LOPRESTI, PH.D.
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF
UNITED STATES PATENT NO. 9,734,825**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	EXPERIENCE, QUALIFICATIONS, AND STATUS AS AN INDEPENDENT EXPERT	2
III.	MATERIALS REVIEWED AND CONSIDERED	12
IV.	UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES.....	12
	A. Anticipation	13
	B. Obviousness.....	13
	C. Priority and Written Description Priority.....	15
V.	TECHNOLOGY OVERVIEW	16
	A. Natural Language Speech Recognition and Spoken Dialogue Systems.....	17
	B. Asynchronous Programming.....	22
	C. Meta-searching	23
VI.	THE '825 PATENT	25
	A. Overview	25
	B. Prosecution History	26
VII.	LEVEL OF SKILL IN THE ART AND PERSPECTIVE APPLIED IN THIS DECLARATION.....	27
VIII.	OVERVIEW OF THE PRIOR ART	29
	A. Ross (Ex. 1005)	29
	B. Ittycheriah (Ex. 1007)	31

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 9,734,825

C.	Hartono (Ex. 1006).....	34
D.	Baker (Ex. 1008)	37
E.	Glover (Ex. 1011).....	42
IX.	CLAIM CONSTRUCTION.....	44
X.	OPINIONS ON GROUND 1.....	44
A.	A POSITA Would Have Been Motivated to Combine	44
B.	A POSITA Would have Had a Reasonable Expectation of Success	47
C.	Independent Claim 5	49
1.	Preamble: “A method for responding to a user generated natural language speech utterance, the method comprising:”	49
2.	Limitation 5[a]: “recognizing, by a speech recognition engine, one or more words in the user generated natural language speech utterance;”	52
3.	Limitation 5[b]: “receiving, at a parser, keyword and associated prior probabilities or fuzzy possibilities from a system agent or an active domain agent of a plurality of autonomous executable domain agents;”	54
4.	Limitation 5[c]: “determining, for the natural language speech utterance, a score for each of at least two possible contexts, wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities;”	59
5.	Limitation 5[d]: “determining by the parser, a domain for the user generated natural language utterance based on the recognized one or more words of the natural language utterance and the determined scores for each of the at least two possible contexts;”	60

6.	Limitation 5[e]: “selecting at least one of the plurality of autonomous executable domain agents based, at least in part, on the determined domain, wherein each of the plurality of domain agents is configured to respond to queries and/or commands within a particular domain, wherein the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands;”	62
7.	Limitation 5[f]: “providing at least one query and/or command based on the natural language utterance to the selected at least one of the plurality of domain agents;”	66
8.	Limitation 5[g]: “creating, by the selected at least one of the plurality of domain agents, one or more queries based on the at least one query and/or command;”	67
9.	Limitation 5[h]: “sending, by the selected at least one of the plurality of domain agents, the one or more queries in an asynchronous manner to one or more local or external information sources.”	68
D.	Dependent Claims 6-7	71
1.	Claim 6: “ <i>The method of claim 5, further comprising generating, by the parser, the at least one query and/or command based on the natural language utterance.</i> ”	71
2.	Claim 7: “ <i>The method of claim 6, wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.</i> ”	72
XI.	OPINIONS ON GROUND 2	73
A.	A POSITA Would Have Been Motivated to Combine	73
B.	A POSITA Would have Had a Reasonable Expectation of Success	76

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 9,734,825

C.	Independent Claim 5	77
1.	Preamble through Limitation 5[f]	77
2.	Limitation 5[g]: “creating, by the selected at least one of the plurality of domain agents, one or more queries based on the at least one query and/or command;”	77
3.	Limitation 5[h]: “sending, by the selected at least one of the plurality of domain agents, the one or more queries in an asynchronous manner to one or more local or external information sources.”	78
D.	Dependent Claims 6-8	80
1.	Claim 6: “ <i>The method of claim 6, wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.</i> ”	80
2.	Claim 7: “ <i>The method of claim 6, wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.</i> ”	80
3.	Claim 8: “ <i>The method of claim 5, wherein one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries and send the duplicate queries in an asynchronous manner to different information sources.</i> ”	80
XII.	OPINIONS ON GROUND 3.....	82
A.	A POSITA Would Have Been Motivated to Combine	82
B.	A POSITA Would have Had a Reasonable Expectation of Success	86
C.	Independent Claim 5	87

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 9,734,825

1. Preamble: “A method for responding to a user generated natural language speech utterance, the method comprising:” 87
2. Limitation 5[a]: “recognizing, by a speech recognition engine, one or more words in the user generated natural language speech utterance;” 88
3. Limitation 5[b]: “receiving, at a parser, keyword and associated prior probabilities or fuzzy possibilities from a system agent or an active domain agent of a plurality of autonomous executable domain agents;” 94
4. Limitation 5[c]: “determining, for the natural language speech utterance, a score for each of at least two possible contexts, wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities;” 96
5. Limitation 5[d]: “determining by the parser, a domain for the user generated natural language utterance based on the recognized one or more words of the natural language utterance and the determined scores for each of the at least two possible contexts;” 101
6. Limitation 5[e]: “selecting at least one of the plurality of autonomous executable domain agents based, at least in part, on the determined domain, wherein each of the plurality of domain agents is configured to respond to queries and/or commands within a particular domain, wherein the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands;” 103
7. Limitation 5[f]: “providing at least one query and/or command based on the natural language utterance to the selected at least one of the plurality of domain agents;” 107

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 9,734,825

8.	Limitation 5[g]: “creating, by the selected at least one of the plurality of domain agents, one or more queries based on the at least one query and/or command;”	108
9.	Limitation 5[h]: “sending, by the selected at least one of the plurality of domain agents, the one or more queries in an asynchronous manner to one or more local or external information sources.”	109
D.	Dependent Claims 6-8	111
1.	Claim 6: “ <i>The method of claim 5, further comprising generating, by the parser, the at least one query and/or command based on the natural language utterance.</i> ”	111
2.	Claim 7: “ <i>The method of claim 6, wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.</i> ”	112
3.	Claim 8: “ <i>The method of claim 5, wherein one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries and send the duplicate queries in an asynchronous manner to different information sources.</i> ”	113
XIII. OPINIONS ON GROUND 4.....		116
A.	A POSITA Would Have Been Motivated to Combine	116
B.	A POSITA Would Have Had A Reasonable Expectation of Success	119
C.	Dependent Claim 8	120
1.	Claim 8: “ <i>wherein one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries and send the duplicate queries in an asynchronous manner to different information sources.</i> ”	120

XIV. SECONDARY CONSIDERATIONS	121
XV. CONCLUSION	121

I. INTRODUCTION

1. I, Daniel Lopresti Ph.D., have been retained by Meta Platforms, Inc. (“Meta”) to provide an analysis of the scope and content of U.S. Patent No. 7,734,825 (“the ’825 patent”) relative to the state of the art at the time of the earliest application to which the ’825 patent claims priority. My analysis relates to claims 5-8. I have also been retained to provide analysis regarding what a person of ordinary skill in the art (“POSITA”) would have understood at the time of the earliest application underlying the ’825 patent.

2. This declaration summarizes the opinions I have formed to date. I reserve the right to modify or amend my opinions, if necessary, based on further review and analysis of information that I receive subsequent to the filing of this declaration, including in response to positions taken by Dialect, LLC (“Dialect”) or its experts that I have not yet seen, including any secondary considerations evidence that Dialect or its expert may consider and present.

3. It is my opinion that claims 5-8 of the ’825 patent are invalid based on the following grounds.

Ground 1	Claims 5-7 are obvious over Ross (Ex. 1005) and Ittycheriah (Ex. 1007)
Ground 2	Claims 5-8 are obvious over Ross (Ex. 1005) and Ittycheriah (Ex. 1007) in view of Glover (Ex. 1011)

Ground 3	Claims 5-8 are obvious over Hartono (Ex. 1006) and Baker (Ex. 1008)
Ground 4	Claim 8 is obvious over Hartono (Ex. 1006), Baker (Ex. 1008), Glover (Ex. 1011)

II. EXPERIENCE, QUALIFICATIONS, AND STATUS AS AN INDEPENDENT EXPERT

4. My curriculum vitae (“CV”) is Exhibit 1004.

5. I am being compensated at the rate of \$500 per hour for my work on this declaration. My fee is not contingent on the outcome of any matter or on any of the technical positions I explain in this declaration.

6. I have no financial interest in Meta. I have been informed that Dialect claims ownership of the ’825 patent. I have no financial interest in Dialect.

7. I am currently a tenured Full Professor in the Department of Computer Science & Engineering (“CSE”) of the P.C. Rossin College of Engineering and Applied Science at Lehigh University in Bethlehem, Pennsylvania and have been since May 2009.

8. I have been a faculty member in the CSE Department since 2003. From July 2009 through June 2019, I also served as Chair of the CSE Department, except for July 2014 to June 2015, during which I served as the Interim Dean of the P.C. Rossin College of Engineering and Applied Science at Lehigh University. In July 2015, I was named Director of the Lehigh Data X strategic initiative focused on

interdisciplinary data science, a role I held until June 2020. From 2003 to 2009, I was an Associate Professor in the CSE Department.

9. I have taught, conducted research, and worked on various aspects of computer science and engineering, including aspects of computer science and engineering related to speech recognition, such as search algorithms used for speech recognition, parallel algorithms, speech synthesis, speech biometrics, and other areas.

10. I received a B.A. in Mathematics and Engineering Science from Dartmouth College in 1982, an M.S.E. in Electrical Engineering and Computer Science from Princeton University in 1983, an M.A. in Electrical Engineering and Computer Science from Princeton University in 1984, and a Ph.D. in Computer Science from Princeton University in 1987.

11. The courses that I have taught at Lehigh include Big Data Analytics, Artificial Intelligence Theory and Practice, AI for Social Good, Software Product Management, Issues in Cybersecurity, Data Mining, Research Methods, Operating System Design, and Breadth of Computing. In my Big Data Analytics, AI, and Data Mining courses I have covered a range of techniques from artificial intelligence and machine learning. The Breadth of Computing course was an introductory course which covered the operation of computer hardware, software, networks, and systems. Topics covered in this course included introductions to designing

websites, writing computer programs, computer architecture, operating systems, computer networks, databases, security, privacy, and social, ethical, and professional issues. As a part of this class, I taught the students HTML and JavaScript programming.

12. From December 2002 to June 2003, I held a courtesy appointment as a Visiting Research Scientist at the Palo Alto Research Center, and from February 2003 to June 2003 I also held courtesy appointments as a Research Fellow in the Information Security Institute at Johns Hopkins University, and as a Visiting Research Associate Professor at Rensselaer Polytechnic Institute.

13. From October 1997 to December 2002, I was a Member of Technical Staff at Bell Labs/Lucent Technologies, conducting research in the areas of computer security, document image analysis, optical character recognition, and pen computing, and where I also collaborated with colleagues on the development of text-to-speech (“TTS”) technologies. I also worked on audio CAPTCHAs, speech biometrics, and handwriting biometrics.

14. From June 1991 to October 1997, I was a Principal Scientist at the Matsushita Information Technology Laboratory in Princeton, NJ, conducting research in the areas of optical character recognition, pen computing, document analysis, and document processing systems.

15. From September 1986 to May 1991, I was an Assistant Professor of Computer Science at Brown University, conducting research in parallel computing and computer-aided design of VLSI. During this time I also taught courses related to VLSI design, parallel computing, discrete math, and computational molecular biology.

16. From 2009 through 2013, I served as Chair of Technical Committee 11 of the International Association for Pattern Recognition (“IAPR”). Technical Committee 11 is known as the “Reading Systems” committee and its goal is to understand and develop systems that can analyze media containing character symbols and return an encoded representation of the text content and structure.

17. From 2018 through 2021, I served as Treasurer of IAPR, and in 2021, I was elected President of IAPR. In 2022, I transitioned to the role of Past-President and held that position until 2024. I served as a member of the IAPR Executive Committee from 2018 through 2024.

18. From 2015 through 2025, I served as an appointed member on the Council for the Computing Community Consortium (“CCC”) of the Computing Research Association (“CRA”). I assumed the role of Vice Chair of CCC effective July 2020 and became Chair of CCC in July 2022. I served in the position of Chair Emeritus of CCC from July 2024 through June 2025 and remain involved in some ongoing CCC and CRA activities. The role of the CCC is to catalyze the United

States computing research community to pursue high-impact research, conduct activities that strengthen the research community, articulate compelling research visions, and align those visions with key national and global challenges.

19. While on the CCC Council, I served on, and also co-chaired, a number of Task Forces, including: Intelligent Infrastructure; Cybersecurity and Cybercrime; Technology for a Connected and Distributed Life; Security, Integrity, and Trust; NextGen AI; and Grand Challenges in Computing. I also served on the Roadmap for Artificial Intelligence Research Steering Committee, which addressed processing of speech and dialog systems, among other AI topics. *A 20-Year Community Roadmap for Artificial Intelligence Research in the United States* (2019), available online at <https://cra.org/ccc/wp-content/uploads/sites/2/2019/08/Community-Roadmap-for-AI-Research.pdf>. I also helped organize the recent CCC Workshop on the Future of Information Retrieval Research in the Age of Generative AI, which was held in Washington, DC in July 2024.

20. Since 2014, I have served as an Editor-in-Chief of the International Journal of Document Analysis and Recognition which is a scholarly journal published by Springer-Nature and generally concerns all areas related to document analysis and recognition. This includes research dealing with computer recognition of characters, symbols, text, lines, graphics, images, handwriting, signatures, as well as automatic analyses of the overall physical and logical structures of documents,

with the ultimate objective of a high-level understanding of their semantic content.

Over a period of more than 20 years, I have served as a technical program co-chair for many of the major international conferences in the fields of document analysis and pattern recognition research.

21. Since 2011, I have also served as an Area Editor for Computer Vision and Image Understanding which is a scholarly journal published by Elsevier. The focus of Computer Vision and Image Understanding is the computer analysis of pictorial information, including all aspects of image analysis from the low-level, iconic processes of early vision to the high-level, symbolic processes of recognition and interpretation.

22. My areas of expertise encompass the fields of document analysis techniques which combine both computer vision and natural language understanding, optical character recognition technology, other artificial intelligence techniques (e.g., machine learning and computer vision), computer and biometric security techniques, electronic voting systems, and research reproducibility.

23. I have authored over 175 publications in the computer science field, including publications relating to the algorithms that are central to the computer processing of speech, including the design of custom computer hardware for parallel algorithms, speech synthesis, speech in security applications, and speech biometrics. I am a named inventor on 24 United States patents. I have authored and co-authored

many publications involving search algorithms and dynamic programming, text-to-speech, computer processing of speech, Hidden Markov Models, and related concepts. My Ph.D. thesis was titled “Discounts for dynamic programming with applications in VLSI processor arrays” (1987), available at <https://www.proquest.com/docview/303617084?pq-origsite=gscholar&fromopenview=true>, which dealt with both the dynamic programming algorithms that are core to speech recognition, and also the parallel implementation of such algorithms.

24. While they are fully set forth in my CV, some examples of my publications include:

- (1) “A Text-to-Speech Platform for Variable Length Optimal Unit Searching Using Perceptual Cost Functions,” M. Lee, D. Lopresti, and J. Olive, *Proceedings of the Fourth ISCA Workshop on Speech Synthesis*, August-September 2001, Perthshire, Scotland, pp. 75-80.
- (2) “Human Interactive Proofs for Spoken Language Interfaces,” D. Lopresti, C. Shih and G. Kochanski, *Proceedings of the Workshop on Human Interactive Proofs*, January 2002, Palo Alto, CA, pp. 30-34.
- (3) “Towards Speech-Generated Cryptographic Keys on Resource-Constrained Devices,” F. Monroe, M. Reiter, Q. Li, D. Lopresti, and C.

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 9,734,825

- Shih, *Proceedings of the Eleventh USENIX Security Symposium*, August 2002, San Francisco, CA, pp. 283-296.
- (4) "A Reverse Turing Test Using Speech," G. Kochanski, D. Lopresti, and C. Shih, *Proceedings of the Seventh International Conference on Spoken Language Processing*, September 2002, Denver, CO, pp. 1357-1360.
- (5) "A Text-to-Speech Platform for Variable Length Optimal Unit Searching Using Perceptual Cost Functions," M. Lee, D. Lopresti, and J. Olive, *International Journal of Speech Technology*, vol. 6, no. 4, October 2003, pp. 347-356.
- (6) "Speech Biometric Mapping for Cryptographic Key Generation," K. Inthavisas and D. Lopresti, *Biometric Technology for Human Identification VIII (SPIE Defense, Security, and Sensing)*, April 2011, Orlando, FL, pp. 80291P-1 - 80291P-12.
- (7) "Attacks on Speech Biometric Authentication," K. Inthavisas and D. Lopresti, *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICCV 2011)*, July 2011, Las Vegas, NV, pp. 310-316.
- (8) "Speech Cryptographic Key Regeneration Based on Password," K. Inthavisas and D. Lopresti, *Proceedings of the International Joint*

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 9,734,825

Conference on Biometrics (IJCB 2011), October 2011, Arlington, VA, 7
pages, DOI: 10.1109/IJCB.2011.6117553.

(9) “Secure Speech Biometric Templates for User Authentication,” K.
Inthavisas and D. Lopresti, *IET Biometrics*, vol. 1, no. 1, pp. 46-54, 2012,
DOI: 10.1049/iet-bmt.2011.0008.

(10) I am a named inventor in 24 United States patents, including:

(11) “Methods and Apparatus for Defending Against Telephone-based
Robotic Attacks Using Random Personal Codes,” H. Baird, J. Bentley, D.
Lopresti, and S.Y. Wang, U.S. Patent No. 7,978,831, issued July 2011.

(12) “Methods and Apparatus for Defending Against Telephone-based
Robotic Attacks Using Contextual-based Degradation,” H. Baird, J.
Bentley, D. Lopresti, and S.Y. Wang, U.S. Patent No. 8,005,197, issued
August 2011.

(13) “Methods and Apparatus for Defending Against Telephone-based
Robotic Attacks Using Permutation of an IVR Menu,” H. Baird, J. Bentley,
D. Lopresti, and S.Y. Wang, U.S. Patent No. 8,005,198, issued August
2011.

(14) I have also been a member of the Institute of Electrical and Electronics
Engineers (“IEEE”) since 1982. I became an IEEE Senior Member in
2003, and an IEEE Life Member in 2024. During my career, I have

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 9,734,825

authored or co-authored approximately 18 refereed papers or proceedings that were published in IEEE technical publications.

- (15) I have also been active in other IEEE activities. For example, I served as program co-chair for the Eleventh International Conference on Document Analysis and Recognition in 2011, whose proceedings were published on the IEEE Explore website. I have received two awards from IEEE organizations. I was an Associate Editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence* between 2005 and 2010. I have been a member of conference program committees for several IEEE conferences, I have served as an invited reviewer for a number of different IEEE journals and several IEEE conferences, all as listed in my CV.
- (16) I am also a member of several other scientific and research associations, and have conducted various activities for them, attended their conferences, and published papers and proceedings in their technical publications. These include the Association for Computing Machinery (“ACM”) and the Association for the Advancement of Artificial Intelligence (“AAAI”).
- (17) In formulating my opinions, I have relied upon my training, knowledge, and experience in the relevant art.

III. MATERIALS REVIEWED AND CONSIDERED

25. My opinions in this declaration are based on working in the field of computer science and engineering, particularly on topics relating to natural language speech recognition, including my teaching, research, and consulting. I have an established understanding of the relevant field at the relevant time frame, and I have an understanding of the skill set, capabilities, and knowledge of a POSITA as of June 2002.

26. My opinions are also based on investigation and study of the patent at issue, its file history, and the prior art. In the course of forming my opinions, I have reviewed the exhibits submitted with the Petition, which are specifically identified in Appendix A.

27. I may rely upon these materials and/or additional materials to rebut arguments raised by Dialect. Further, I may also consider additional documents and information in formulating any necessary opinions including documents that may not yet have been provided to me.

IV. UNDERSTANDING OF RELEVANT LEGAL PRINCIPLES

28. I understand that statutory and judicially created standards must be considered to determine the validity of a patent claim. I am not an attorney and, consequently, will offer no opinion on the law itself. My understanding of the

pertinent law is described in this section and is the result of explanations provided by counsel. I have applied this understanding in my analysis.

29. I understand that a patent claim is unpatentable if it is obvious in view of the prior art. I further understand that the frame of reference for determining whether a claim is obvious is from the perspective of a POSITA at the time of invention. I have been asked to assume that the effective priority date is June 3, 2002. I have not been asked to determine whether this date is appropriate.

A. Anticipation

30. I have been informed that for a patent claim to be anticipated by the prior art, each and every limitation of the claim must be found, expressly or inherently, in a single prior art reference as recited in the claim. I have been informed that a claim limitation not expressly found in a prior art references is inherent if the prior art necessarily functions in accordance with, or includes, the claim limitation. Mere probability that a limitation is included is not sufficient to establish inherency.

B. Obviousness

31. In analyzing obviousness in light of the prior art I have been informed that it is important to understand the scope of the claims, the level of skill in the relevant art, the scope and content of the prior art, the differences between the prior

art and the claims, and any objective indicia of non-obviousness (also called secondary considerations).

32. I have been informed that a patent claim is unpatentable for obviousness if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA to which said subject matter pertains. I have been informed that obviousness may be based on one reference or a combination of references. I have been informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.

33. I have been informed that when a patented invention is a combination of known elements, the Board must determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue by considering the teachings of prior art references, the effects of demands known to people working in a field or present in the marketplace, and the background knowledge possessed by a POSITA.

34. I have been informed that the Supreme Court has recognized several rationales for combining references or modifying a reference to show obviousness of the claimed subject matter. I understand that several of these rationales are: (1) combining prior art elements according to known methods to yield predictable

results; (2) simple substitution of one known element for another to obtain predictable results; (3) use of a known technique to improve a similar device (method or product) in the same way; (4) applying a known technique to a known device (method or product) ready for improvement to yield predictable results; (5) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; (6) and some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

35. I have been informed that in order to prove that a claimed invention is not patentable for obviousness, a petitioner must (1) identify the differences between the claim and particular disclosures in the prior art references, singly or in combination, (2) specifically explain how the prior art references could have been combined in order to arrive at the subject matter of the claimed invention, and (3) specifically explain why a POSITA would have had reasons to so combine the prior art references.

C. Priority and Written Description Priority

36. I have been informed that to claim priority to a prior application, each application in the chain that leads to an earlier application must comply with the written description requirement. I have been informed that in order to satisfy the written description requirement, the specification of the provisional application must

contain a written description of the invention and the manner and process of making and using it, in such full, clear, concise, and exact terms, that it would enable a person of ordinary skill in the art to practice the invention claimed in the non-provisional application. I have been informed that the provisional application must describe the invention claimed in the non-provisional application such that a person of skill in the art can clearly conclude that the inventor had invented the claimed invention by the provisional application filing date.

V. TECHNOLOGY OVERVIEW

37. In this section I provide overviews of technologies and prior art pertinent to the '825 patent and to the state of the art as of 2002. In later sections, I address in detail how the prior art discloses and renders obvious the recitations of the challenged claims.

38. The prior art references described below and discussions of what was known to a person of ordinary skill in the art provide a general description of the state of the art at the time of the invention, provide additional motivation to modify the primary references evaluated in the grounds of invalidity with the knowledge of a person of ordinary skill in the art or other references cited herein, and provide support for why a person of ordinary skill in the art would have had a reasonable expectation of success in combining the teachings of the references. The additional

prior art references identified in this section are exemplary in nature and are included to provide factual support for my opinions.

A. Natural Language Speech Recognition and Spoken Dialogue Systems

39. Natural language speech recognition refers to the process in which a computer system converts spoken language into written text. This enables interaction and communication between human and computers through speech and voice. Natural language speech recognition methods have been used for decades. By the mid-1990s, several companies including IBM and Dragon Systems, had already been selling speech recognition systems off-the-shelf as ways to process user-generated natural language speech into machine readable text to be used by various applications. *See generally* Ex. 1017 (IBM ViaVoice 1998 User Manual), Ex. 1018 (IBM ViaVoice 2002 User Manual), Ex. 1023 (Dragon Systems Naturally Speaking 5 User's Guide).

40. These speech recognition systems used statistical probabilities, language modeling, and context-recognition procedures to recognize speech. By 2002, one of the most common acoustic models for speech recognition that used probabilities was a concept referred to as the hidden Markov model ("HMM"). Ex. 1019, 22. As Federick Jelenick, a well-known researcher in the field of speech recognition wrote, "[a] *Markov chain* is a very important and well-studied concept of probability theory. It deals with a class of random processes that incorporate a

minimum amount of memory without actually being memoryless.” Ex. 1019, 22. Essentially, HMMs apply probabilities to recognize words based on how they transition between one acoustic data point to the next and the probability of a specific transition.

41. By 2001, IBM’s ViaVoice product was already incorporated into an IBM “Personal Speech Assistant,” which was a “complete speech recognition, speech synthesis, and dialog management system.” Ex. 1021, 1, 3. It employed HHM states modeled by a Gaussian mixture to look at the likelihood between different transitions to recognize phrases. Ex. 1021, 3. Using these statistical probabilities, a user could speak into the Personal Speech Assistant, which would be received by a dialogue manager (DM). Ex. 1021, 4. After speech recognition is performed, the DM would convert the utterance into events using a vocabulary stored in a database. Ex. 1021, 2, 4. The DM would then send the “associated event and data” to “the target application.” Ex. 1021, 4. For example, natural language utterance around appointments could be sent to a speech-enabled date book application, such that queries or commands for new appointments “can be transcribed directly into the calendar as text[.]” Ex. 1021, 4.

42. Importantly, they also routinely used dictionaries and phrase tables as reference vocabularies to identify words. As Frederick Jelenick also noted, “[e]very

speech recognizer is based upon a vocabulary of finite size.” Ex. 1019, 62. These “vocabularies” contain lists of words and phrases, that are often specific to a speaker:

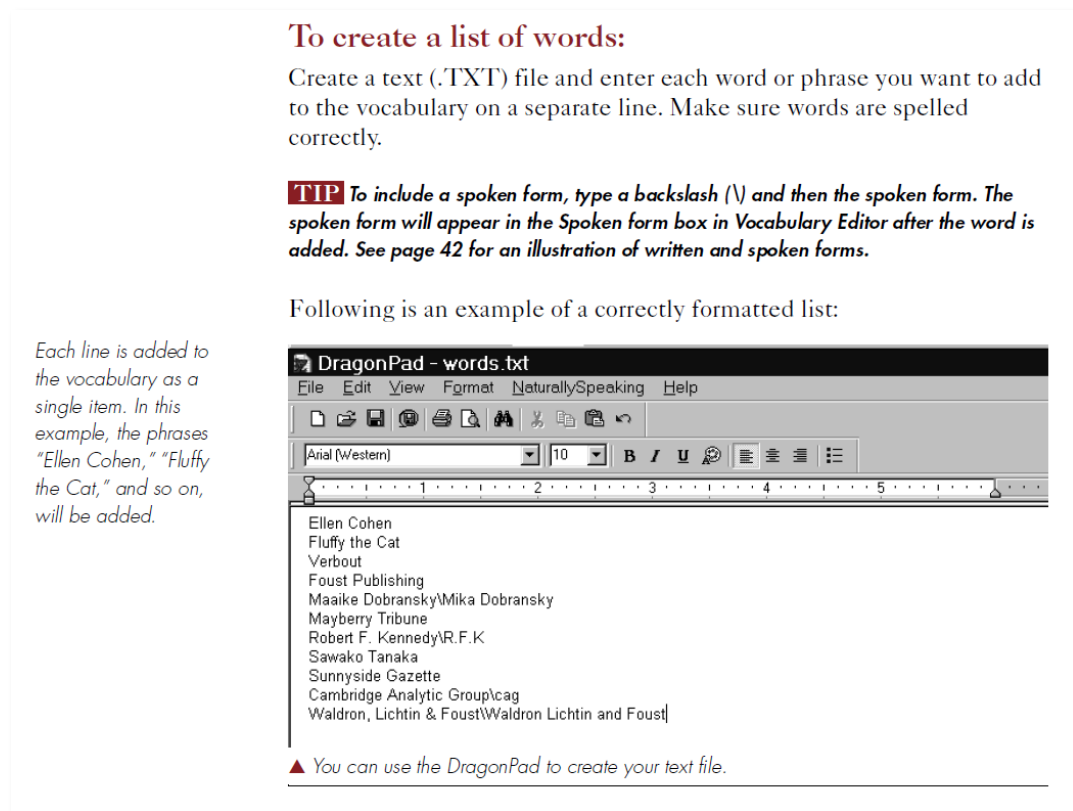
A vocabulary is particular to a speaker and his task in at least four different aspects. First, choice of words reflects the speaker’s habits of expression that are related, for example, to his level of education. His usage is also conditioned by the general domain of discourse (e.g., data processing, musicology, medical reports, etc.) that calls for a variety of technical terms, clichés, and such. Furthermore, the speaker has available those names, addresses, and other expressions peculiar to his current interests. Finally, he requires words specific to the topic of the particular document he is composing.

A large, fixed vocabulary can at most provide for the speaker’s habitual usage and for the discourse domain, but a completely satisfactory coverage can be attained only through a user’s active, though limited, participation. In particular, although he can be asked to indicate whether a proper noun is a personal or company name, location, etc., and even to annotate technical terms, he cannot be expected to provide any grammatical feature classification of ordinary English words. Information about the latter should either be contained in some large stored source lexicon, or, better still, be extracted from the actual syntactic/semantic function that these words play in the previously transcribed text.

Ex. 1019, 63. These vocabularies were often dynamic in nature, as a way to personalize the vocabulary to the speaker. *Id.* (“A crude method of personalization could be devised by maintaining during dictation a dynamically varying vocabulary consisting at any given moment of the last L different words used. The resulting *dynamic coverage* is then defined as the asymptotic probability that the word spoken next already belongs to the vocabulary.”). In other words, a vocabulary built from the speakers most recently used words would be the most probable words the speaker

would use again in follow-on spoken inputs. Ex. 1023, at 47-65 (describing the dynamic vocabulary used by Dragon Systems that is customized by a user either manually or through adding previously recognized words to a vocabulary).

43. In the below example, it is shown how a vocabulary consists of both individual words and phrases:



Ex. 1023, 57.

44. By 2002, there were several different types of spoken dialogue systems that allowed users to interact with computer-based applications, including agent-based systems. See Ex. 1020, 3. Agent-based systems in particular were designed to allow for more complex communications between the system, user, and

underlying application. Such systems would often use various different knowledge sources to process spoken dialogue including a dialogue history between the system and a domain model, with specific information about the domain. Ex. 1020, 6. An agent-based system is a broad concept, but it generally includes interactions between two agents, “each of which is capable of reasoning about its own actions and beliefs, and sometimes also about the actions and beliefs of the other agent.” Ex. 1020, 5.

45. For example, an application itself can act as an agent if it interacts with another system agent—like conversation or dialogue manager—to respond to a user’s spoken utterance. Such systems can be seen in U.S. Patent No. 7,127,402, in which computer programs themselves are “agents”:

In the approach of the present invention, the speech center system **20** has an explicit model of the world (e.g., domain model **70**) which will serve as a foundation for language understanding and reasoning. Some of the basic concepts that the speech center system **20** models using the domain model **70** are:

Things	A basic category that includes all others	15
Agents	Animate objects, people, organizations, computer programs	
Objects	Inanimate objects, including documents and their sub-objects	
Locations	Places in the world, within the computer, the network, and within documents	20
Time	Includes dates, as well as time of day.	
Actions	Things that agents can do to alter the state of the world	
Attributes	Characteristics of things, such as color, author, etc.	
Events	An action that has occurred, will occur, or is occurring over a span of time.	

Ex. 1012, 7:8-25

Ex. 1012, 8:12-16 (“The speech center 20 has a rudimentary built-in notion of what an ‘action’ is. An ‘action’ is something that an agent can do in order to achieve some change in the state of the word (e.g., known to the speech center 20 and an application 26).”)

B. Asynchronous Programming

46. In the field of computing (in particular, software engineering), it is generally assumed that when there are multiple applications present in a system they will behave asynchronously. This means that each application will execute at its own rate, independent of the execution of other applications running on the same system. Ex. 1024, 3 (“asynchronous” defined as “[P]ertaining to, being, or characteristic of something that is not dependent on timing. For example, asynchronous communications can start and stop at any time instead of having to match the timing governed by a clock” (Microsoft Computer Dictionary 4th Ed. (1999))).

47. The alternative – synchronous execution – actually requires substantially more program code and effort to implement, as the applications must be designed to communicate with each other at various phases during their execution to stay in sync, and if one application gets too far ahead of the other it will be paused (“blocked”) until the other application has had a chance to catch up. Such coordination can be complex and potentially error-prone, so it is avoided when it is

not necessary for the correct operation of the applications. We can tell that applications execute asynchronously when there is no indication of, or need for, step-by-step coordination between the applications. An application that simply returns its final result when it is done is still operating asynchronously.

C. Meta-searching

48. In its early days, the Internet experienced explosive growth. It quickly became impossible for a user to benefit from the totality of content available online that was relevant to a question they might have or a task they wanted to perform. More and more search engines were developed to help users navigate an increasingly complex information landscape. To anyone who tried the various alternatives available for search, it was apparent that different search engines would produce different results, even for an identical query. No single search engine dominated all the others: depending on a user's specific information needs, one search engine might be more effective than its competitors, but it would not always be the same one. While it was possible to conduct a search by manually invoking several engines and browsing through the results from each, this clearly placed a significant extra burden on the user.

49. The concept of meta-search was created to automate the process of querying multiple search engines and integrating their results in a way that produced more a comprehensive response to a user's information needs. Using a meta-search

engine, a user asks just one query and the engine determines which other search engines to engage. Because the secondary search engines are stand-alone systems fielded by different, competing vendors, they run autonomously. The meta-search engine receives the results from those searches and integrates them before returning a response to the user. Going beyond simply reporting all of the hits returned, a meta-search engine could combine and reformulate the results, recognizing that information repeatedly returned by different search engines might be more authoritative and valuable to the user. If one search engine misses an important resource – it has a “blind spot” – that will not hurt the user because other search engines are likely to know about that resource. A meta-search engine could also employ the output from one search engine to refine its query to another, allowing it to investigate a topic more thoroughly. In all these ways, a meta-search engine acts as an agent working on behalf of the user to seek out and aggregate as much information as possible for the benefit of the user.

50. In the case of a speech interface, meta-search offers even more benefits because it can produce a more comprehensive response to a query while requiring less back-and-forth interaction between the user and the system.

VI. THE '825 PATENT

A. Overview

51. I understand that the '825 patent claims priority to a provisional application U.S. Provisional Application No. 60/384,388 filed on June 3, 2002.

52. The '825 patent describes well-known systems and methods for natural language processing of user speech. In particular, the '825 patent uses well-known system and methods for speech recognition to recognize words spoken by the user, prior probabilities or fuzzy possibilities to determine possible contexts for the words, and then determines actions based on the context.

53. The '825 patent acknowledges that “[m]uch work covering multiple methods has been done in the fields of natural language processing and speech recognition,” and “[s]peech recognition has steadily improved in accuracy and today is successfully used in a wide range of applications.” *Id.*, 1:56-60. The '825 patent further recognizes that “[n]atural language processing has previously been applied to the parsing of speech queries.” *Id.*, 1:60-61.

54. The '825 patent claims recite a combination of these well-known systems and methods. Generally, the '825 patent recites the steps of using a “speech recognition engine” to recognize one or more words in user-generated speech. *Id.*, 27:9-28. The patent then recites using the one or more words with associated prior probabilities or fuzzy possibilities to determine possible contexts for the received

utterance and provides a query or command based on that context. *Id.*, 27:28-28:11.

The '825 patent lastly recites sending the query or command to a local or external information source to be executed. *Id.*, 28:11-17. To perform several of these steps, the '825 patent teaches using an “active domain agent of a plurality of autonomous executable domain agents,” which it describes as being “complete, convenient and re-distributable packages or modules for each application area,” i.e., “everything needed to extend or modify the functionality of the system. *Id.*, 11:58-63.

B. Prosecution History

55. I understand that the original claims of the '825 patent were repeatedly rejected over the prior art. In particular, I understand that the Examiner commented that “The Applicants’ current claims are simply too broad given what is known in this field of the prior art.” Ex. 1002, 220.

56. On March 22, 2017, I understand that the Applicant amended the claims to include the following limitation: “wherein each of the selected at least one of the plurality of domain agents is configured to create one or more queries based on the at least one query and/or command and send the one or more queries in an asynchronous manner to one or more local or external information sources.” *Id.*, 231.

57. I understand that the Applicant stated that the prior art “describes agents as being able to issue request to facilitator 600 for additional assistance” but “does

not disclose or suggest that the requests are sent in an asynchronous manner.” *Id.*,

235. But as I discuss above in the Technology Overview, asynchronous programming is the general default for software. *See* Section V.B (Asynchronous Programming).

58. On April 12, 2017, I understand that the Examiner allowed the claims, stating that the prior art of record does “not disclose or reasonably suggest sending queries in an asynchronous manner.” Ex. 1002, 253.

VII. LEVEL OF SKILL IN THE ART AND PERSPECTIVE APPLIED IN THIS DECLARATION

59. I understand that certain issues relating to validity must be judged from the perspective of a person of ordinary skill in the relevant art, as I discussed below.

60. The purported invention of the ’825 patent generally relates to well-known systems and methods of processing natural language speech.

61. In determining the level of ordinary skill, I have been asked to consider, for example, the types of problems encountered in the field, prior solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology, and the educational level of active workers in the field. In my opinion, it would have taken at least two years of experience in the field for a person to become familiar with the problems encountered in the field and the prior and current solutions to those problems.

62. Taking those factors into consideration, a person of ordinary skill in the art (“POSITA”) in June 2002 (i.e., the earliest possible priority date of the ’825 patent) would have had at least a bachelor’s degree in electrical engineering, computer science, computer engineering, or an equivalent, and two years of relevant experience involving computer science fundamentals, including natural language processing, and speech recognition and transcription. Additional experience could substitute for less education, and additional education could likewise substitute for less experience.

63. My opinions do not turn on the precise definition, and the challenged claim would be unpatentable from the perspective of any reasonable POSITA.

64. I believe that I would have qualified as a person of at least ordinary skill in the art as of the earliest claimed June 3, 2002 priority date on the face of the ’825 patent. I believe that I have a sufficient level of knowledge, experience, and education to provide an expert opinion in the field of the ’825 patent.

65. For the purpose of this declaration, and unless otherwise noted, my statements and opinions, such as those regarding my experience and understanding of a POSITA generally (and specifically related to the references I consulted) reflect the knowledge that existed in the field as of June 3, 2002. This is true even if stated in the present tense.

VIII. OVERVIEW OF THE PRIOR ART

A. Ross (Ex. 1005)

66. Ross is a U.S. patent assigned to IBM entitled “System and Method for Determining Utterance Context in Multi-Context Speech Application” filed on August 16, 2001 and issued on August 1, 2006. Ross discloses each limitation of the system recited in independent claim 5 except for the details of the *speech engine*. It was well-known at the time that IBM was developing technology in the natural language speech processing field.

67. Ross describes a system that includes a “speech center” that “coordinates speech services for a number of speech-enabled applications performing on a computer system,” including the “task of deciding the target application for a particular utterance by a user.” Ex. 1005, Abstract, 2:3-9, Fig. 1.

68. When the system receives a speech input from the user, Ross uses a speech engine to convert spoken commands (i.e., requests) “into text strings that can be utilized by software applications,” i.e., a “speech representation.” Ex. 1005, 6:62-7:2. Ross recognizes that by 2001, speech engines were already well known and commercially available including ViaVoice™ from IBM and NaturallySpeaking Professional from Dragon Systems. Ex. 1005, 1:16-22; *see generally* Ex. 1017, Ex. 1018, Ex. 1025.

69. Ross describes a “context manager” that receives the speech representation from such a “speech engine” and proceeds to “select a context 70 for a speech enabled application 26 by checking the speech representation against contexts 70.” Ex. 1005, 7:19-23. Ross ranks the contexts list based on an “access characteristic,” such as the “recency of relevant access to the context,” as shown below in Figure 4:

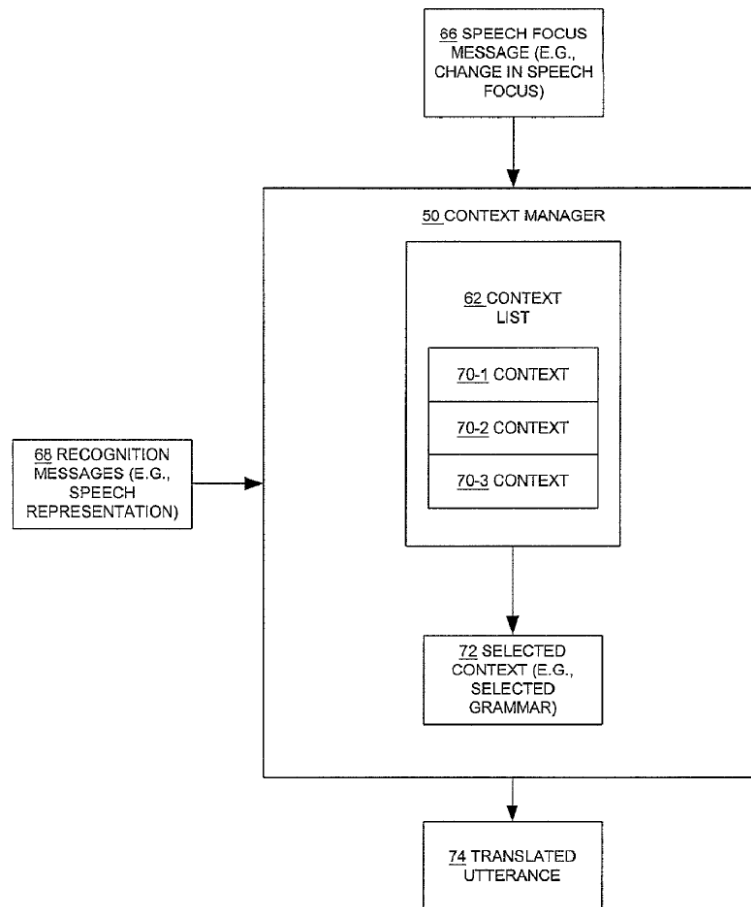


FIG. 4

Ex. 1005, Fig. 4, 2:47-49, 3:1-3, 6:47-61, 8:46-9:13

70. In this way, Ross maintains a set of prior probabilities of possible contexts “biased to send commands to the most recently accessed application,” that is continuously being “reordered.” Ex. 1005, 8:47-53; *see also* 5:58-61.

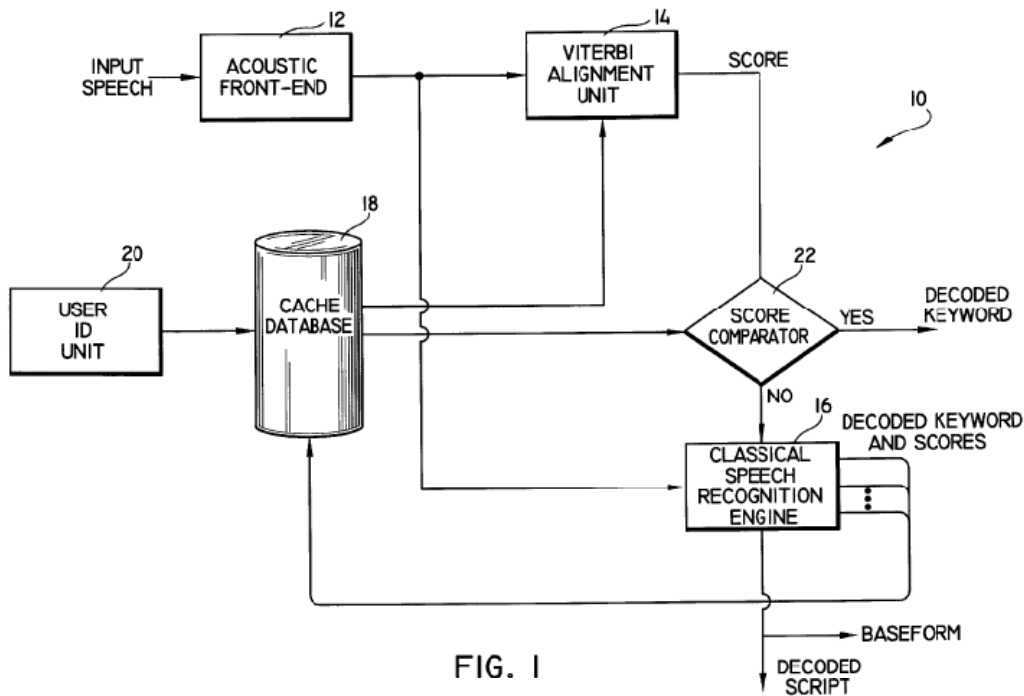
71. Ross then uses “the grammar associated with each context 70 to determine if the [speech representation] is capable of being processed by the grammar,” and selects the context based on this matching (i.e., relevance scoring). Ex. 1005, 7:23-26.

72. Ross translates the speech representation “according to the selected context 72 and directs the translation to the script engine,” the script engine then runs the script against the target application, thereby invoking it to perform the function of the command and present the results to the user. Ex. 1005, 7:27-28.

B. Ittycheriah (Ex. 1007)

73. Ittycheriah is a U.S. patent assigned to IBM entitled “Apparatus and Methods for Speech Recognition Including Individual or Speaker Class Dependent Decoding History Caches for Fast Word Acceptance or Rejection” filed on June 4, 1997 and issued on August 10, 1999. Ittycheriah discloses the details of the *speech engine* of independent claim 5.

74. Ittycheriah discloses a speech recognition engine for “performing speech recognition on speech segments frequently input by a user,” as shown below in Figure 1:



Ex. 1007, Fig. 1, Abstract, 1:11-15, 1:51-54

75. Ittycheriah achieves this by “storing data in a cache and utilizing such data in order to quickly accept or reject keywords, i.e., names, command, or sentences, without impacting the overall recognition process and, therefore, without adversely affecting recognition performance.” Ex. 1007, 2:59-63.

76. Ittycheriah’s cache database “contains the word history of a keyword (name, command or sentence) as well as the likelihood (score) obtained when the word was added to the cache.” Ex. 1007, 2:63-66.

77. Ittycheriah uses a set of prior probabilities and possibilities to perform speech recognition, explaining that once speech is input into its system:

[T]he acoustic features extracted in acoustic front-end 12 are subject to probabilistic matching technique using Hidden Markov models (HMMs) in both the alignment unit 14 and the engine 16... It is to be understood that Hidden Markov modeling provides different scores for different hypotheses. Examples of such scores are rank (i.e., the ranking of a decoded word), fast match scores (i.e., likelihoods of a simplified scheme), detailed match scores (i.e., the likelihoods of a detailed scheme), language modeling scores (i.e., probability according to a language model) and/or any combination of these scores.

Ex. 1007, 3:56-4:4.

78. Ittycheriah's cache database compares the "resulting scores" to "scores which were previously stored in the cache database 18 pertaining to the particular user are presented to the score comparator 22 wherein the scores from unit 14 are compared to the stored scores." Ex. 1007, 4:33-43.

79. "If the new scores are within a margin of, the same as, or larger than the stored scores corresponding to one of the decoded keywords in the cache, then the keyword is accepted," and the decoded keyword in the cache is outputted from the score comparator. Ex. 1007, 4:43-48.

80. If the new scores are "not within a margin of error, or are less than any of the stored scores, then the decoded keywords in the cache are rejected and the

input speech segment is subjected to the complete vocabulary of the classical speech recognition engine,” from which a decoded script (i.e., a keyword) is outputted. Ex. 1007, 4:48-53. The decoded keyword and associated scores from the classical speech engine are then added to the cache database. Ex. 1007, 4:54-57.

81. Ittycheriah’s vocabulary in its “classical speech engine” (and usable by its cache database) contains a “larger scale language database.” Ex. 1007, 3:62-64.

82. The vocabulary stores “keywords,” which Ittycheriah defines as “words and phrases as names, commands, and sentences, which include both functional and non-functional words,” or “other forms of vocabulary.” Ex. 1007, 1:55-58, 2:16-19, 2:47-49.

83. A POSITA would thus have understood that Ittycheriah’s vocabulary would have been considered a “dictionary” of words and “phrase tables,” including commands and sentences in the same way as the ’825 patent.

C. Hartono (Ex. 1006)

84. Hartono is a PCT application entitled “Adaptive Natural Language Interface,” with an International Publication date of March 2, 2000. Ex. 1006, Cover. Hartono discloses each limitation of the system recited in independent claim 5 except for the details of the *speech engine*.

85. Hartono discloses an agent-based system of processing natural language utterances. Hartono discloses a “method for receiving commands,

executing received commands and adaptively interacting with the user using a natural language interface.” Ex. 1006, 1:20-23.

86. To that end, Hartono describes a “natural language agent adapted to receive and interpret natural language user input and to output a command,” and an “application agent adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program. Ex. 1006, 4:20-25.

87. Hartono illustrates its use of various application agents in Figure 1, below:

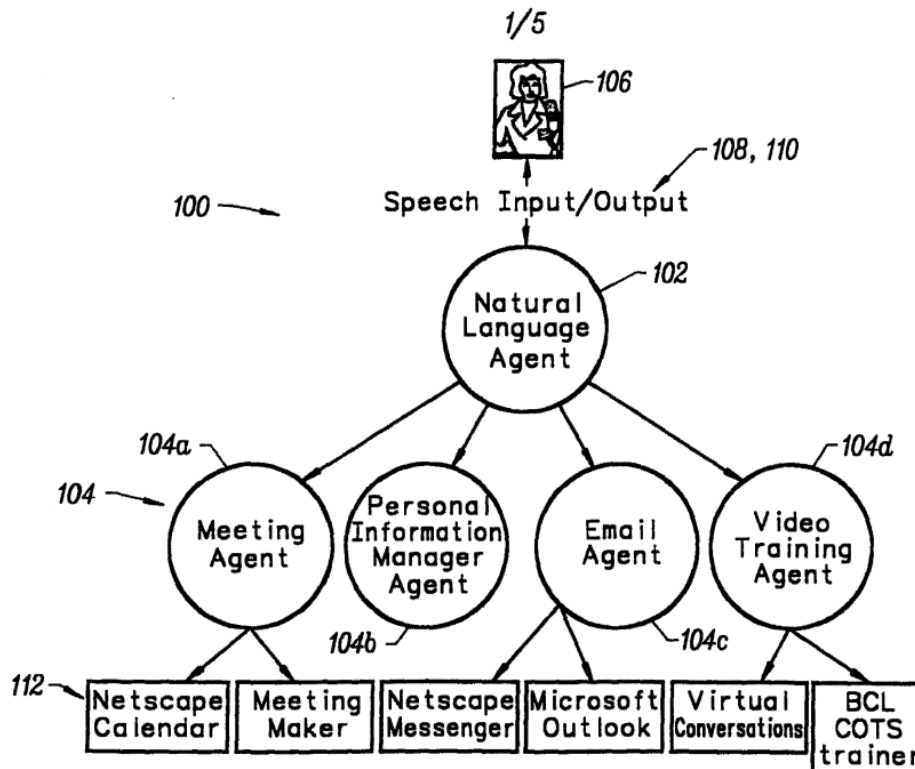


FIG. 1

Ex. 1006, Fig. 1, 7:6-9

88. Hartono initially processes input speech using “[a]ny suitable off the shelf speech recognition system.” Ex. 1006, 11:4-12.

89. After receiving the recognized speech, Hartono discloses using a “natural language syntactic parser” to “parse natural language computer commands” from the recognized, for example, converting “the verb into a computer command action, with the noun phrase (NP) as the object and the adjective phrase (AP) as the properties of the object.” Ex. 1006, 12:11-13:10.

90. Hartono then takes the parsed natural language and “maps it into a generic concept frame used to invoke the appropriate application method,” as shown in Table I:

TABLE I		
ACTION-CONCEPT	TOPIC CONCEPT	APPLICATION-CONTEXT
Show	Email	Email application
Show	Email address	Address book application
Delete	Email	Email application
Show	General help	Natural language agent

Ex. 1006, Table I

91. To complete this mapping, Hartono’s semantic interpretation engine uses a “reverse grammar generation mechanism” that “includes a list or vector for each word and corresponding probabilities for each word in the list.” Ex. 1006, 14:10-20.

92. For example, Hartono describes the words “I,” “eye,” or “aye,” having probabilities of 80%, 15%, and 5%, respectively such that the word with the highest probability is most likely to be suitable for the given context. Ex. 1006, 15:4-12.

93. Using both the action-concept (verb) and topic-concepts (noun/object), Hartono determines the context of the natural language request by “determin[ing] which task agent should handle the request.” Ex.1006, 15:13-19.

94. Hartono communicates with various task agents (also called “application class agents”), which “works with a single class of applications...that have similar conceptual operations.” Ex. 1006, 23:15-23, 28:1-3 (“Each task agent thus serves as the common user interface for the corresponding class of applications under the assumption that each application within the class accomplishes the same or generally overlapping set of tasks.”).

95. Based on the determine context, Hartono “invokes a routine that sends a set of requests to the task application wrapper,” which is “written in one of the platform specific macro languages.” Ex. 1006, 24:14-25:3.

96. The command within the natural language request is then executed by the selected application. Ex. 1006, 4:20-25, 5:4-8, 5:11-15, 29:21-22, 42:11-12.

D. Baker (Ex. 1008)

97. Baker is a U.S. patent assigned to Dragon Systems entitled “Systems and Methods for Word Recognition” filed on June 7, 1995 and issued on October

21, 1997. Dragon Systems was a well-known developer of speech recognition software in the mid-to-late 1990s.

98. Baker describes systems and methods for “detect[ing] words within portions of speech signals and compar[ing] these detected words to acoustic models of known words, or sequence of acoustic models of known words.” Ex. 1008, 18:13-16.

99. Upon detection of a “speech signal,” Baker compares the detected speech to “acoustic models of known words, or sequence of acoustic models of known words.” Ex. 1008, 18:12-16.

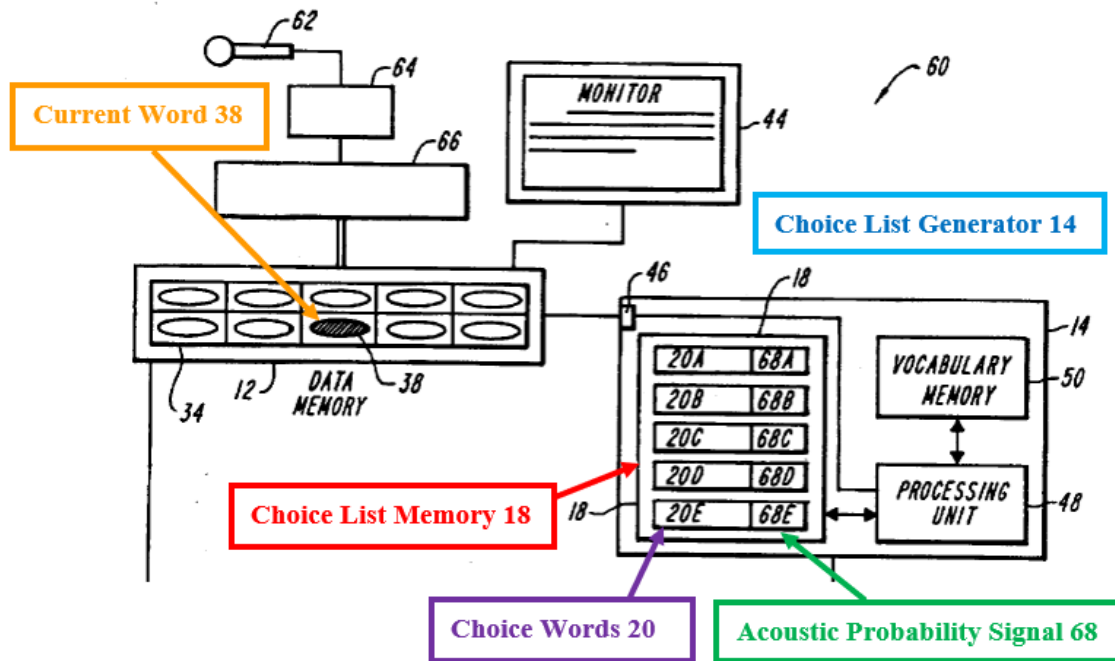
100. “This comparison leads to a probability signal that indicates the certainty that the detected word is substantially acoustically similar to the word associated with the acoustic model.” Ex. 1008, 18:16-19.

101. Words with a probability signal indicating a “high degree of certainty” are deemed recognized (referred to as “context words”). Ex. 1008, 18:16-19.

102. However, words with a “low degree of certainty” indicate that the word has not been confidently identified (referred to as the “current word”). Ex. 1008, 18:22-24.

103. For those “unrecognized words,” Baker maintains a “choice list 18” that “store[s] the words 20 associated with the acoustic models from the vocabulary

memory 50 which are acoustically similar to the current ambiguous word 38 being identified, as illustrated below in the following excerpt of Figure 2:



Ex. 1008, Fig. 2 (excerpted and annotated)

104. Baker explains that each “choice word 20A-E” has a corresponding “acoustic probability signals 68A-E” that represent “the degree of certainty that the word 20 associated with the acoustic model is substantially representative of the current word 38 being recognized.” Ex. 1008, 18:30-35, 20:30-46 (“The probability signals 68A-68E can represent the likelihood that the associated word choice 20A-20E is substantially representative of the current word 38 being recognized.”), 22:55-64.

105. Baker also discloses generating a “word list” of words that “have a semantic association with the choice word 20.” Ex. 1008, 14:57-69.

106. The “semantic association” for each choice word (which Baker also refers to as “passage words” (labeled as item 24 in Fig. 2) in the “word list” is derived from a “reference database 40” which can be a “dictionary, thesaurus, antonym dictionary, medical dictionary, legal dictionary, scientific and engineering dictionary, or other type of reference source.” Ex. 1008, 20:55-58 (“The word list generator 16 can access via a transmission path the reference database 40 and can identify information in the reference database 40 that relate to the topic, theme, or subject of a choice word 20.”).

107. Baker next “perform[s] a matching function to calculate the number of matched words” between those words recognized by the speech analysis unit 66 with high confidence (the “context words”) with the words in the “word list” (the “passage words”). Ex. 1008, 21:19-47.

108. By looking at the “relative frequency of matches between the “context words” and the “passage words,” the Baker calculates a “probability signal 28 representative of the likelihood that the choice word or phrase 20 substantially represents the utterance currently being recognized.” Ex. 1008, 21:19-47, 21:49-22:6, Figs. 2-4.

109. Baker then generates a “ranking signal 30” that “represents the likelihood...that a choice word 20 is substantially similar to” the word being recognized” as a function of both the “context matching and acoustic model matching” probabilities. Ex. 1008, 23:22-27.

110. Baker selects the choice word with the highest rank signal, and supplants the unrecognized word in the text string with the choice word as a way of recognizing the received utterance. Ex. 1008, 12:54-62, 17:23-27 (“[T]he word recognition system 10 can select the choice word 20 with the highest probability signal 28 or rank signal 30 to substitute for the current word 38 currently being recognized.”).

111. Baker recognizes that by 1997 it was conventional for natural language speech recognition systems to “rely on data intensive language models to identify” words, and these models are essentially statistical probability models that include a set of hypotheses as to what words are likely to appear in association with other words.” Ex. 1008, 2:1-12, 2:36-47 (discussing the n-gram language model to identify similar sequences of words), 17:11-15 (“However, it will be apparent to one of ordinary skill in the art of word recognition that many known techniques can be employed to generate the probability signal 28. Any of these methods, or variations thereof, can be used without departing from the scope of the present invention.”), 21:65-22:6 (“Techniques for generating the probability signal given two

constraints...are well known in the field of mathematics and any of the techniques capable of generating a probability signal given a set of constraints, including the maximum entropy technique, can be practiced with the present invention without departing from the scope thereof.”).

E. Glover (Ex. 1011)

112. Glover is a U.S. patent application publication entitled “Selective Retrieval Metasearch Engine” filed on June 29, 2001, and published on November 7, 2002.

113. Glover discloses a metasearch engine, which “takes user queries and submits them to multiple underlying search engines, and combines the results into a single interface.” Ex. 1011, ¶¶35, 47 (“A retrieve search engine results step 52 for sending queries to search engines or databases and retrieving the results from the search engines or databases[.]”); *see id.*, ¶3 (“A metasearch engine accepts a search query, and send the query (possibly transformed) to one or more regular search engines, and collects and processes the responses from the regular search engines in order to present a list of documents to the user.”).

114. Glover’s system is illustrated below in Figure 2:

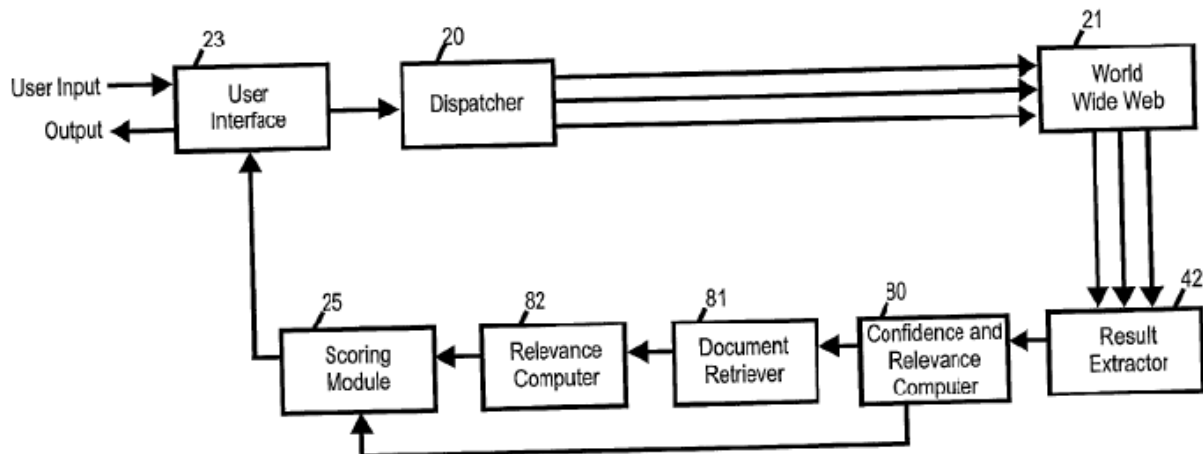


FIG. 8

Ex. 1011, Fig. 8; *see also id.*, Fig. 2, ¶¶36-38.

115. Relevant here, to facilitate fast, accurate searching, Glover describes using a “dispatcher” (item 20) to “make[] the primary search decisions for a metasearch engine,” including “which search engines to query, and how to query each source.” Ex. 1011, ¶¶38, 39 (“The dispatcher includes a source selector 31 to choose search engines to query generator 32 to modify queries appropriately for each source.”), Fig. 3.

116. Glover explains that “some of the steps” of its claimed method “may be performed in a parallel,” including “sending queries 51 to and retrieving results 52 from search engines.” Ex. 1011, ¶82.

IX. CLAIM CONSTRUCTION

117. I have applied the plain meaning of the terms and phrases in the claims in my analysis, as a POSITA would have understood those terms and phrases at the relevant time (around 2002). I understand that the Board will construe the claims using the same claim construction standard that would be used in district courts. My declaration will establish that the prior art meets each of the claim limitations under any reasonable construction. I have not been asked to consider any express constructions.

X. OPINIONS ON GROUND 1

118. In my opinion, claims 5-7 are rendered obvious by Ross (Ex. 1005) in view of Ittycheriah (Ex. 1007).

A. A POSITA Would Have Been Motivated to Combine

119. In my opinion, a person of ordinary skill in the art would have been motivated implement the details of the Ross “speech engine” using Ittycheriah’s speech recognition method because Ross teaches its speech engine could use any off-the-shelf speech recognition system, including those available from IBM, like the Ittycheriah system.

120. Ross discloses that its “speech center,” shown below as item 20 in Figure 2, is distinct from the “speech engine,” which is used to “convert speech into

text strings that can be utilized by software applications on a computer.” Ex. 1005, 1:7-22.

121. Although Ross does not disclose the details of the “speech engine,” Ross teaches that off-the-shelf speech engines could be used to accomplish this step. For example, Ross teaches that “[i]n a preferred embodiment, the speech engine 22 is ViaVoice™ from IBM®.” Ex. 1005, 4:37-38; *see also* 1:16-21 (“Examples of speech recognition products that convert speech into text strings that can be utilized by software applications on a computer system include the ViaVoice™ from IBM®, Armonk, N.Y., and NaturallySpeaking Professional from Dragon Systems, Newton, Mass.”). Indeed, the provisional patent application from which Ross depends, states that “[w]hile we certainly expect to make use of IBM’s ViaVoice speech engine, we would be able to easily replace it with a better alternative if one came along.”

122. Accordingly, a person of ordinary skill in the art would have been motivated to look for and rely on available speech recognition methods to use as Ross’s “speech engine,” including Ittycheriah’s speech recognition engine which was also developed by IBM.

123. A person of ordinary skill in the art would have also been motivated to implement the Ross speech engine using Ittycheriah as a way to quickly complete the speech recognition process by way of its cache database of recently used keywords and probability scores. At the time, there were multiple design choices to

improve the accuracy and speed of speech recognition, and utilizing recently used keywords and probability scores were among those known design choices. These features would have advanced the objectives of Ross, which includes an objective that “avoids the time, expense, and footprint of duplicating speech services.” Ex. 1005, 2:14-16.

124. In addition, a person of ordinary skill in the art would have been motivated to use Ittycheriah’s speech recognition system as Ross’s speech engine because Ittycheriah is able to complete its speech recognition faster than classical speech engines by way of its cache database that stores recently-recognized keywords and associated probabilities. *See supra*, Section VIII.B (Ittycheriah Overview).

125. Because it is desirable for speech-enabled requests and command (like those inputted into Ross) to be processed quickly, a person of ordinary skill in the art would have found it desirable to use Ittycheriah’s cache database to “reduce the amount of hypotheses to consider in the decoding process.” Ex. 1007, 2:13-19, 2:58-63 (“Generally, the present invention provides for storing data in a cache and utilizing such data in order to quickly accept or reject keywords, i.e., names, commands or sentences, without impacting the overall recognition process and therefore, without adversely affecting recognition performance.”).

B. A POSITA Would have Had a Reasonable Expectation of Success

126. In my opinion, a person of ordinary skill in the art would have had a reasonable expectation of success in combining the teachings of Ross and Ittycheriah.

127. Ross teaches that its speech engine 22, as shown below in Figure 2, could be any commercially-available speech recognition engine. Accordingly, a person of ordinary skill in the art would have used the software algorithms of Ittycheriah's speech recognition process to implement the details of speech engine 22:

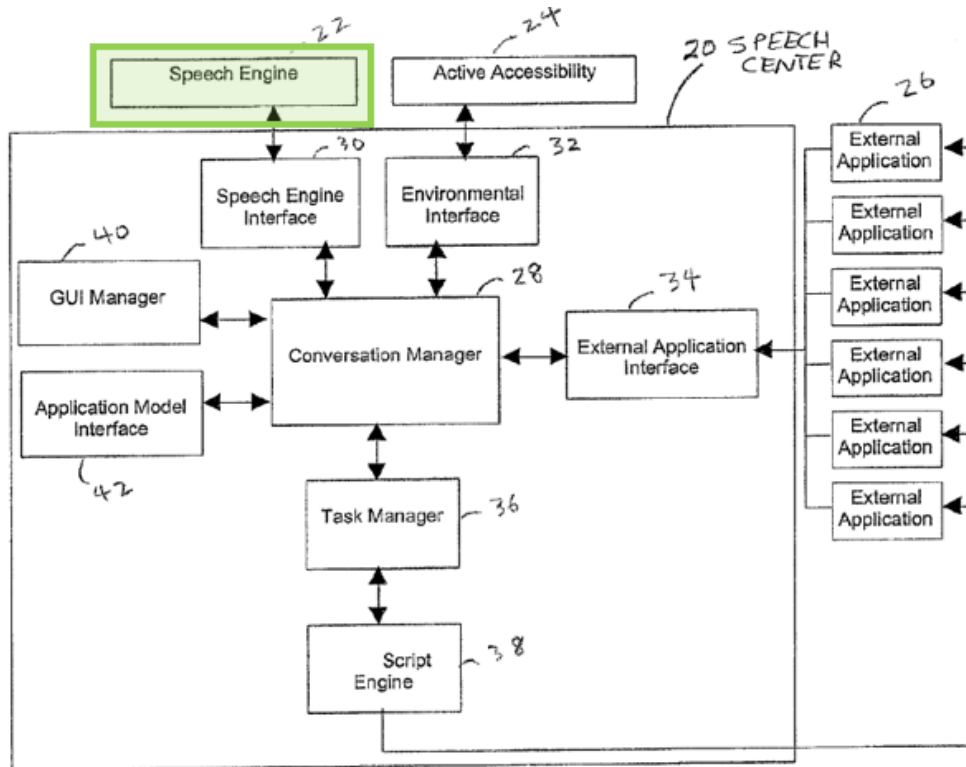


Fig. 2

Ex. 1005, Fig. 2 (annotated)

This is particularly true since both Ross and Ittycheriah were developed by IBM.

128. A POSITA would have also understood that Ittycheriah's speech recognition system would be compatible with Ross, as Ittycheriah discloses that the system's decoded keywords and scripts could be used to direct speech requests to external speech-enabled applications to perform their respective automatic functions, just like in Ross. Ex. 1007, 4:63-5:3 ("It is to be appreciated that the output of the comparator 22 (decoded keyword) and the output of the engine 16 (decoded script) may be connected to the actual phone system or navigation system

[speech recognition applications] such that the phone or navigation systems are responsive to the decoded keyword or script and thus perform the respective automatic functions (i.e., dialing or navigating).”).

129. A POSITA would have also had the skill to implement and the reasonable expectation of success in achieving such a combination because it would have involved a combination of known technologies (e.g., known systems for speech recognition and processing natural language requests) according to known methods (e.g., known methods of speech recognition engines that convert speech into text strings that can be used by software applications) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke various speech-enabled applications on a computer.

C. Independent Claim 5

130. In my opinion, Ross in view of Ittycheriah renders obvious independent claim 5.

1. Preamble: “A method for responding to a user generated natural language speech utterance, the method comprising:”

131. In my opinion, Ross discloses the preamble to the extent limiting.

132. As discussed in Section VIII.A (Ross Overview), Ross discloses a method for “determining a speech-enabled application to receive a spoken utterance in a multi-context speech enabled environment.” Ex. 1005, 2:43-46, Abstract.

133. The spoken utterance is *a user generated natural language speech utterance* and the speech-enabled application is for *responding to the utterance*. application is for *responding to the utterance*. Ross discloses a “speech center” that is responsive to *user generated natural language speech*. Ex. 1005, 5:34-38.

134. Ross discloses several examples of its methods responding to natural language requests. For example, “[i]f the speech center **20** hears a phrase such as ‘print the first message’ or ‘print the first appointment,’ the context manager **50** can readily figure out the intended target application **26** for the uttered sentence.” Ex. 1005, 8:25-33.

135. As another example, Ross describes that when “the user asks whether the user has any new messages,” the user “receives an answer supplied by a message or mail application.” Ex. 1005, 2:25-31 (“In another example, the user asks what is the current stock price for a particular company and obtains the stock price from stock monitoring service.”), 2:31-36 (“[T]he user makes a request to take a letter, and a word processing application starts up (i.e., is launched by the speech center) if it is not already running.”).

136. Natural language refers to the use of word or phrases in human-understandable language. Thus, a person of ordinary skill in the art would have understood that the voice commands discussed by Ross are *natural language* because they use phrases in human-understandable language.

137. Ross discloses *responding to a user generated natural language speech utterance* as illustrated below in Figure 1, which shows receipt by a processor of “input spoken utterance 14,” which is a “voice command or other speech input from a user of a computer system”:

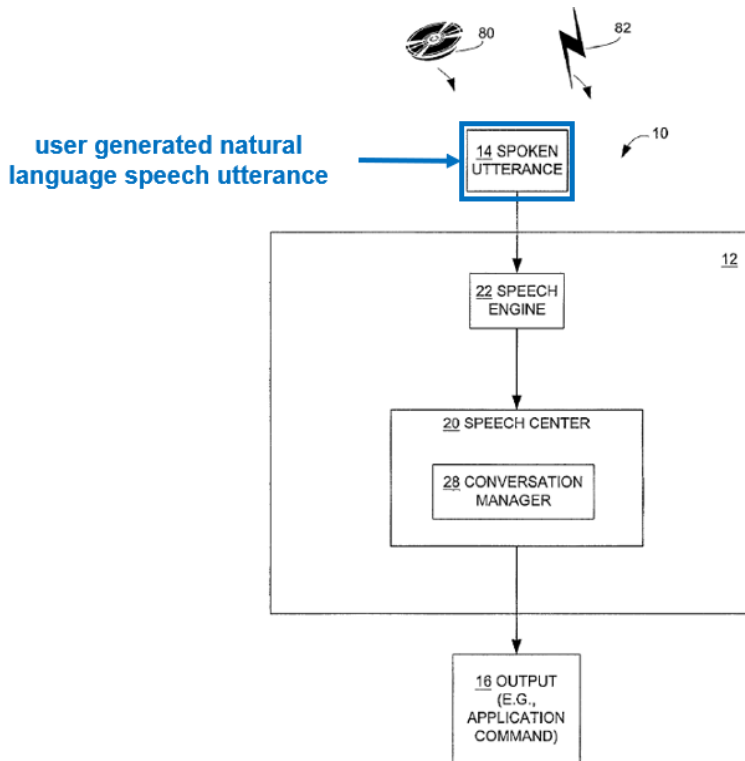


FIG. 1

Ex. 1005, Fig. 1 (annotated), 3:41-53

138. The method is responsive to the user-generated natural language speech because it produced an output 16, which is “a command or other input based on the spoken utterance 14 directed to the speech enabled external application 26.” Ex. 1005, 3:41-53.

139. The external application then presents to the user an answer to the spoken utterance. Ex. 1005, 2:22-36.

2. Limitation 5[a]: “recognizing, by a speech recognition engine, one or more words in the user generated natural language speech utterance;”

140. In my opinion Ross in view of Ittycheriah renders obvious this limitation.

141. As I discussed for the preamble, Ross discloses a method that receives a spoken utterance (*user generated natural language speech utterance*) and responds to that spoken utterance using the speech center to provide an output. Ex. 1005, 5:34-38.

142. The speech center receives text strings provided by a “speech engine.” Ross discloses that the “speech engine” recognizes words in the spoken utterance as shown below in Figure 1.

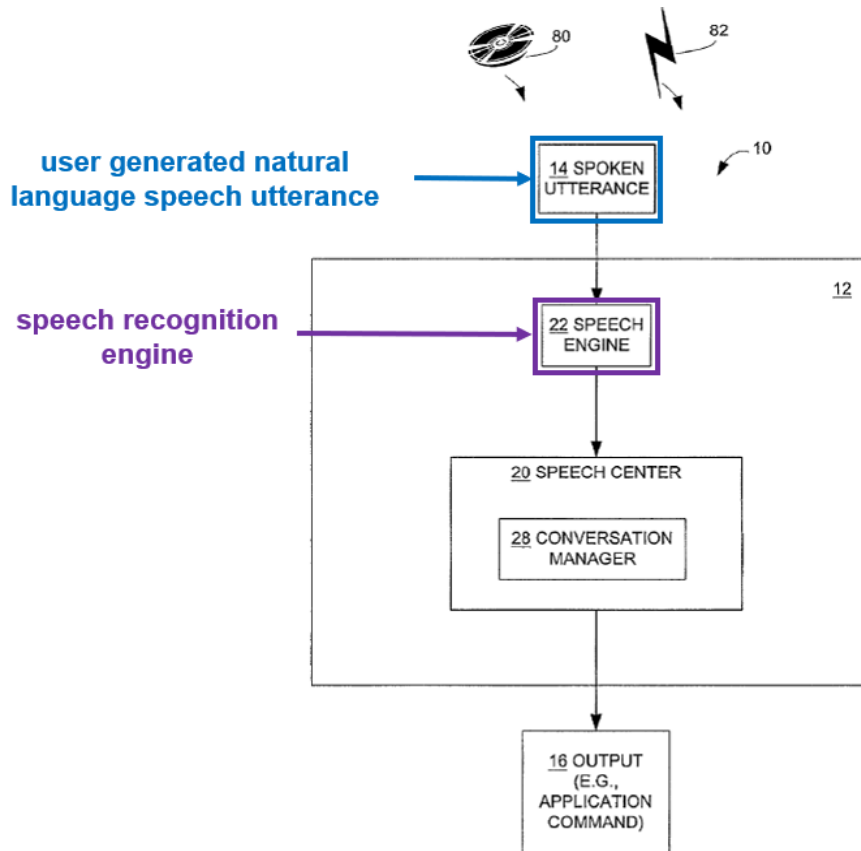


FIG. 1

Ex. 1005, Fig. 1 (annotated)

143. Ross does not disclose the details of the speech engine. Instead, Ross teaches that any well-known “speech recognition products” could “convert speech into text strings,” such as the ViaVoice™ from IBM® and NaturallySpeaking Professional from Dragon Systems. Ex. 1005, 1:17-22.

144. Ittycheriah teaches such a speech recognition product also developed by IBM®. Ex. 1007, 1:10-15 (“The present invention relates to speech recognition

systems.”), cover at [73] (identifying assignee “International Business Machines Corporation”).

145. The Ittycheriah system recognizes one or more keywords (*recognizing, by a speech engine, one or more words*) in a speech segment, such as the spoken utterance of Ross (*in the user generated natural language speech utterance*) and outputs a “recognized script (name, command or sentence).” Ex. 1007, 3:2-6.

146. A person of ordinary skill in the art would have used Ittycheriah’s speech recognition system as the speech engine in Ross given that both systems were developed by IBM and Ross expressly taught a person of ordinary skill in the art to use available speech recognition products as the speech engine, such as that in Ittycheriah. *See* Section X.A (Motivation to Combine) and X.B (Reasonable Expectation of Success).

3. Limitation 5[b]: “receiving, at a parser, keyword and associated prior probabilities or fuzzy possibilities from a system agent or an active domain agent of a plurality of autonomous executable domain agents;”

147. In my opinion, Ross in view of Ittycheriah renders obvious this limitation.

148. As a threshold, the ’825 patent explains that a *parser* “transforms [recognized words and phrases] into complete commands and questions.” Ex. 1001, 11:19-21. This is consistent with the understanding of a person of ordinary skill in the art. Ross’s “speech center” includes a *parser* as part of its “context manager,”

because it determines a context for the received keyword that transforms it to recognized text that has meaning. *See also* Section X.C.4 (Ground 1, Limitation 5[c]) and X.C.5 (Ground 1, Limitation 5[d]).

149. Further, the '825 patent explains that a *system agent* “provides default functionality and basic services.” Ex. 1001, 11:57-58. Ittycheriah’s speech recognition is a *system agent* because it is a component of the Ross system that provides the basic service of recognizing words in the speech utterance and outputs a decoded keyword (*keyword*) along with its likelihood score (*associated prior probabilities or fuzzy possibilities*).

150. Specifically, Ittycheriah discloses a method for “performing speech recognition on speech segments frequently input by a user.” Ex. 1007, Abstract. Ittycheriah leverages frequent inputs by using a cache database.

151. As part of its speech recognition method, Ittycheriah discloses “storing data in a cache database,” in which the cache “contains the word history of a keyword (name, command or sentence) as well as the likelihood (score) obtained when the word was added to the cache.” Ex. 1007, 2:63-66.

152. As Ittycheriah explains, the “likelihood score” is created as part of the “Viterbi alignment unit,” using a “probabilistic pattern matching technique using Hidden Markov Models (HMMs)” which “provid[es] different scores for different hypotheses.” Ex. 1007, 3:54-60. Ittycheriah provides different examples of how

those “likelihood scores” are defined using prior probabilities: (1) by “rank (i.e., the ranking of a decoded word),” (2) by “fast match scores (i.e., the likelihoods of a simplified scheme), (3) by “detailed match scores (i.e., the likelihoods of a detailed scheme),” (4) by language modeling scores (i.e., the probability according to a language model),” and “any combination of these scores.” Ex. 1007, 3:64-4:4.

153. By June 2002, HMMs were well-known and ubiquitous in the field of speech recognition. *See* Ex. 1019 (Jelinek, Frederick, *Statistical Methods for Speech Recognition*, The MIT Press (1997)), 15 (“A *Markov chain* is a very important and well-studied concept of probability theory. It deals with a class of random processes that incorporate a minimum amount of memory without actually being memoryless.”).

154. The Hidden Markov Model uses probability distributions to determine the maximum likely sequence of transitions between a starting state of an input and the probability of a given output. *See id.*, 17-25 (describing how transitions sequences based on probability distributions are calculated by HMMs).

155. Therefore, Ittycheriah’s cache database that stores “likelihood scores” are *associated prior probabilities or fuzzy possibilities* of decoded keywords.

156. Ittycheriah describes its cache database as being created from and constantly updated by way of the classical speech recognition engine, labeled as item 16 below in Figure 1:

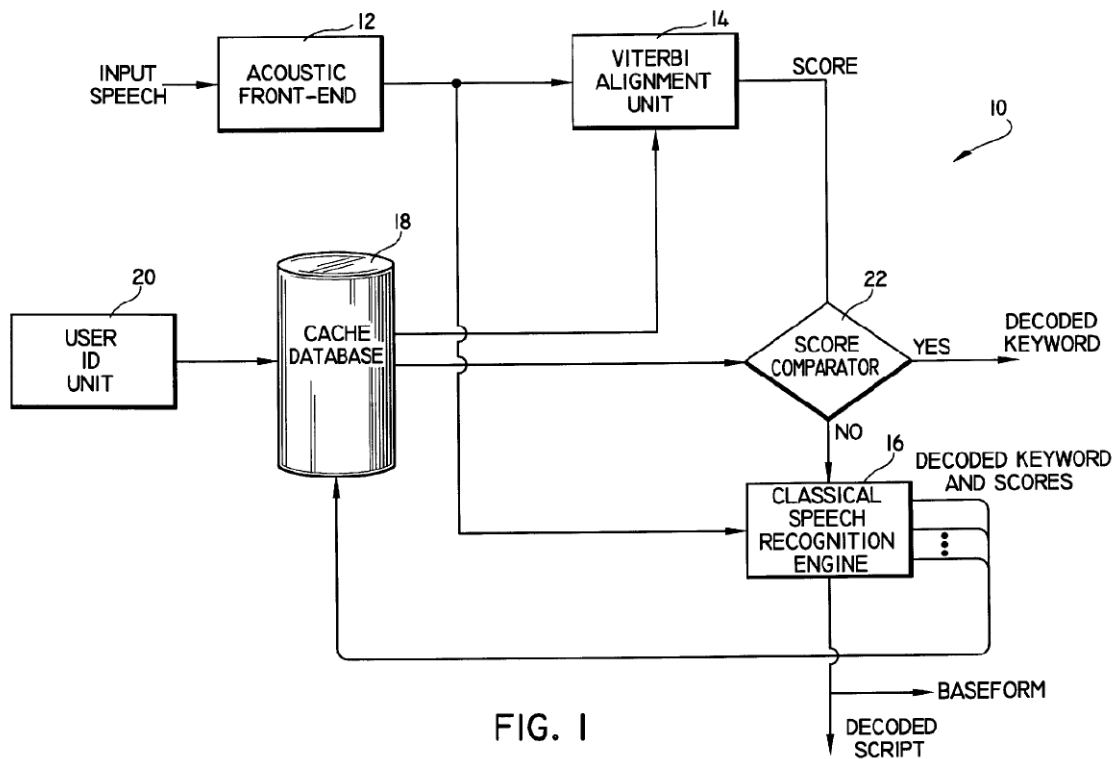


FIG. 1

Ex. 1007, Fig. 1; *id.* at 2:66-3:18, 4:33-59; *see id.* at 1:59-2:10

157. As Ittycheriah explains, once the method receives the input speech (corresponding to the “received utterance”), it applies a probabilistic pattern matching process via the Viterbi alignment unit 14 to assign various likelihood scores to input speech. Ex. 1007, 4:33-39.

158. The resulting scores are then “presented to the score comparator 22.” Ex. 1007, 4:38-40.

159. The “score comparator” then compares the new scores relating to input speech to the scores of decoded keywords that “were previously stored in the cache database.” Ex. 1007, 4:40-43.

160. As Baker explains “[i]f the new scores are not within a margin, or are less than any of the stored scores [in the cache database], then the decoded keywords in the cache are rejected and the input speech segment is subjected to the complete vocabulary of the classical speech recognition engine 16 whereby a decoded script is generated.” Ex. 1007, 4:48-53.

161. Baker further teaches that: “After the input speech is decoded by the engine 16, the decoded word and associated scores are provided to the cache database 18 where they are stored as new keyword and related scores for that particular user. In this way the cache stores the latest and most frequently used keywords that a user may input into the system.” Ex. 1007, 4:57-59; 3:16-18 (“Subsequently, the decoded word(s) resulting from the recognition performed by the engine is then preferably added to the cache along with associated scores.”); 8:3-5 (claim 22).

162. In the combination, Ittycheriah’s “decoded keyword” corresponds to the “recognition message” in Ross, which is output by the speech engine because it is a “speech representation recognized by the speech engine 22.” Ex. 1005, 7:1-2; Ex. 1007, 4:63-53 (“It is to be appreciated that the output of the comparator 22 (decoded keyword) and the output of the engine 16 (decoded script) may be connected to [external applications] such that the [external applications] are

responsive to the decoded keyword or script and thus perform their respective automatic functions.”).

163. A POSITA would have found it obvious to use Ittycheriah’s “decoded keyword” (*keyword and associated prior probabilities or fuzzy possibilities*) as a recognition message in Ross, which is *received* by Ross’s context manager (*parser*). *See also* Section X.C.4 (Ground 1, 5[c]) and X.C.5 (Ground 1, 5[d]).

4. Limitation 5[c]: “determining, for the natural language speech utterance, a score for each of at least two possible contexts, wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities;”

164. In my opinion, Ross in view of Ittycheriah renders obvious this limitation.

165. As discussed for limitation 5[a], Ittycheriah acts as the speech engine in Ross that is used to recognize words and phrases contained in the received utterance and outputs a “recognition message” to Ross’s context manager. Ex. 1005, 6:12-15, 7:12; *see also* Section X.C.2 (Ground 1, Limitation 5[a]).

166. Based on that recognition message (*the received keyword and associated prior probabilities*), Ross discloses *determining...a score for each of at least two possible contexts*.

167. Ross *scores* those possible contexts based on the received keyword using the recency of access characteristic (a *relevance score*).

168. For example, Ross states that the priority order of context grammars in its context list (such as context list 62) is maintained based on the recency of access characteristic of the context grammars in a context list. Ex. 1005, 6:7-9 (“The context manager 50 maintains the priority and state of the various grammars in the context list 62 in the system.”); Ex. 1005, 7:17-21; Figs. 4, 5.

169. Ross’s “context manager” then “tests the utterance against these grammars (indicated by the contexts 70 in the context list 62) in priority order, and passes the commands on to the first application 26 that has a grammar that will accept the phrase.” Ex. 1005, 8:41-46.

170. In other words, based on the received speech recognition message (the *keyword and associated prior probabilities and fuzzy possibilities*), Ross scores the possible contexts within its context list using recency of access such that “the system 20 is biased to send the commands to the most recently accessed application 26.” Ex. 1005, 8:54-9:6.

5. Limitation 5[d]: “determining by the parser, a domain for the user generated natural language utterance based on the recognized one or more words of the natural language utterance and the determined scores for each of the at least two possible contexts;”

171. In my opinion, Ross discloses this limitation.

172. Ross describes how its context manager (*parser*) *determines a domain for the user generated natural language utterance based on the recognized one or more words and the determined score.*

173. Ross gives an example based on multiple ambiguous “print it” or “print this” requests to mail and calendar applications:

The user opens the first message in the electronic mail application 26, which results in the context 70 for the electronic mail application 26 being placed first in the context list 62. If the user says “print it,” then the first message is printed.

The user then opens the second appointment in the calendar application 26, which results in the context 70 for the calendar application 26 being placed first in the context list 62. If the user says “print this,” then the second appointment is printed.

The user next opens the third message in the electronic mail application 26, which results in the context 70 for the electronic mail application 26 being placed first in the context list 62.

The user then uses the mouse to click on the fourth appointment displayed in a window for the calendar application 26. As a result, the context 70 for the calendar application 26 supersedes the context 70 for the electronic mail application 26 in the context list 62. If the user says “print it,” then the fourth appointment is printed.

Ex. 1005, 8:54-9:6.

174. Accordingly, Ross has identified *at least two possible contexts*—a calendar application and an electronic mail application. Ex. 1005, 8:54-9:6.

175. Ross's context manager *determines a domain for the user generated natural language utterance* (either the calendar application or electronic mail) based upon *the recognized word* “print it” or “print this” and based on the relevancy of access characteristic (*the determined score*). Ex. 1005, 9:7-20.

176. As Ross explains, “the context manager 50 resolves the ambiguous phrase (e.g., “print this” or “print it”) by looking in the context list 62 to determine the top priority context 70. Then assuming the grammar for the top priority context 70 can process (i.e., recognize) the phrase, the context manager 50 directs a translation of the phrase to the application 26 indicated by the top priority context 70.” Ex. 1005, 9:7-20, Fig. 5.

6. **Limitation 5[e]: “selecting at least one of the plurality of autonomous executable domain agents based, at least in part, on the determined domain, wherein each of the plurality of domain agents is configured to respond to queries and/or commands within a particular domain, wherein the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands;”**

177. In my opinion, Ross discloses this limitation.

178. Ross discloses *one of the plurality of autonomous executable domain agents*.

179. The '825 patent describes “domain agents” as “autonomous executables that receive process, and respond to user questions, queries, and commands,” that “provide complete, convenient and redistributable packages or modules of functionality.” Ex. 1001, 2:50-59, 2:41-42 (“Any application that supports a robust automation user interface can participate.”).

180. A POSITA would thus understand a “domain agent” as claimed to be an autonomous executable software module specific to each application area.

181. Ross discloses a *plurality* of “external applications 26” that “communicate with the speech center **20**, a speech engine **22**, and active accessibility module **24**.” Ex. 1005, 4:16-21.

182. Each of these external applications have a defined grammar, “along with associated translations for a script engine **38**.” Ex. 1005, 6:29-31.

183. And each of these external applications *is configured to respond to queries and/or commands within a particular domain*, as they are “speech-enabled applications” that respond to speech. Ex. 1001, Abstract, 2:22-36.

184. As Ross explains:

The speech center can allow cross-application speech. In one example, a user is working with a spreadsheet application, but then asks about the user’s next appointment and obtains the information from a calendar program. In another example, the user asks what is the current stock price for a particular company and obtains the stock price from a stock

monitoring service. Furthermore, the user asks whether the user has any new messages and receives an answer supplied by a message or mail application.

Applications do not even have to be running (i.e., currently performing on a computer system). For example, the user makes a request to take a letter, and a word processing application starts up (i.e., is launched by the speech center) if it is not already running.

Ex. 1005, 2:22-36

185. In the same vein, Ross discloses a task manager that “provides the capability to proceed with multiple execution requests simultaneously, to queue up additional script commands for busy applications 26, and to track the progress of execution.” Ex. 1005, 5:1-5.

186. Based on these disclosures in Ross, a POSITA would have thus understood that the external applications in Ross are autonomous executables triggered by the script commands issued by Ross’s task manager.

187. Accordingly, each of Ross’s external, speech-enabled applications is a *domain agent* as claimed in the ’825 patent.

188. Ross further discloses *selecting* the domain agent *at least in part, based on the determined domain*.

189. Specifically, Ross “maintains a context list of speech enabled applications” that the user has accessed and teaches “deciding the target application

(i.e., *selecting a domain agent*) for a received utterance based in part on the determined context (i.e., the *determined domain*). For example, Ross looks at the syntactic parse of the received utterance and evaluates whether that parse is accepted by the grammar of a specific speech-enabled external application. Ex. 1005, 8:33-36.

190. If so, Ross teaches that the “associated application 26 is the one that should be targeted to receive the corresponding command.” Ex. 1005, 8:33-36.

191. Ross describes how *the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands*.

192. For example, Ross describes how requests regarding appointments and dates (*the particular domain*) is an area of expertise for a calendar application (*the domain agent*). Ex. 1005, 2:22-25.

193. As another example, Ross describes how stocks (*the particular domain*) is an area of expertise for a stock monitoring service (*the domain agent*). Ex. 1005, 2:25-28, 2:28-30 (“[T]he user asks whether the user has any new messages and receives an answer supplied by a message or mail application.”), 2:32-36 (“For example, the user makes a request to take a letter, and a word processing application starts up (i.e., is launched by the speech center) if it is not already running.”).

194. Accordingly, a POSITA would understand that Ross discloses this limitation based on the many examples disclosed.

7. Limitation 5[f]: “providing at least one query and/or command based on the natural language utterance to the selected at least one of the plurality of domain agents;”

195. In my opinion, Ross discloses this limitation.

196. Ross teaches that the natural language utterance contains *at least one query and/or command*.

197. Ross discloses that “[t]he input spoken utterance 14 is a voice command or other speech input from a user of the computer system 10 (e.g., into a microphone connected to the computer system 10). Ex. 1005, 3:45-48.

198. A POSITA would understand that a “voice command” is *at least one query and/or command*. Ex. 1005, 3:45-48, 9:40-43 (“[T]he spoken utterance including a command that is able to be associated with multiple applications, any subset of which may or may not be running[.]”), 2:22-36 (describing various utterances such asking about the user’s “next appointment” or “current stock price for a particular company,” each containing “at least one request”).

199. Ross further discloses *providing the at least one query and/or command to the selected at least one of the plurality of domain agents*.

200. After selecting the target external application (i.e., *domain agent*) as discussed in 5[e], Ross translates the received utterance to be processed by the

external application and calls and dispatches the request to the target external application (*at least one of the plurality of domain agents*). See Ex. 1005, 5:55-57 (“The utterance is then translated into the appropriate script engine 38 calls and dispatched to the target application 26), 6:15-18 (“When a successful match is found, the corresponding translation 74 is dispatched to the script engine 38 for execution[.]”), 6:48-51 (“The first application’s grammar which will accept the utterance is then used for translation, and the command dispatched to the corresponding application 26), 7:26-29 (“In step 108, the context manager 50 direct the speech representation to be translated according to the selected context 27[.]”), 9:9-13, Abstract, Fig. 5 (step 108).

201. This translation is associated with the “grammar defined for each application.” Ex. 1005, 6:29-31 (“Toward that end, a grammar is defined for each application 26, along with the associated translations for a script engine.”).

8. Limitation 5[g]: “creating, by the selected at least one of the plurality of domain agents, one or more queries based on the at least one query and/or command;”

202. In my opinion, Ross discloses this limitation.

203. Ross’s external applications (*one of the plurality of domain agents*) *create...one or more queries based on the at least one query and/or command.*

204. After selecting the target external application (i.e., domain agent), Ross translates the received utterance to be processed by the external application. Ex.

1005, 5:55-57 (“The utterance is then translated into the appropriate script engine 38 calls and dispatched to the target application 26), 6:15-18 (“When a successful match is found, the corresponding translation 74 is dispatched to the script engine 38 for execution[.]”), 6:48-51 (“The first application’s grammar which will accept the utterance is then used for translation, and the command dispatched to the corresponding application 26), 7:26-29 (“In step 108, the context manager 50 direct the speech representation to be translated according to the selected context 27[.]”), 9:9-13, Abstract, Fig. 5 (step 108).

205. This translation is associated with the “grammar defined for each application.” Ex. 1005, 6:29-31 (“Toward that end, a grammar is defined for each application 26, along with the associated translations for a script engine.”). A POSITA would have understood Ross’s “translation” of the received utterance would be creating a query based on the at least one natural language query or command. *E.g.*, Ex. 1005, 6:48-51 (“The first application’s grammar which will accept the utterance is then used for translation, and the command dispatched to the corresponding application 26).

9. Limitation 5[h]: “sending, by the selected at least one of the plurality of domain agents, the one or more queries in an asynchronous manner to one or more local or external information sources.”

206. In my opinion, Ross discloses this limitation.

207. A POSITA would have understood that the term “asynchronous” in the programming context includes sending the queries to be executed autonomously, rather than being controlled by the main programming flow.

208. Asynchronous programming is a well-known paradigm of computer programming that allows for different tasks to be performed independently of the other. *See* Section V.B (Technology Overview).

209. The '825 patent also describes sending “asynchronous” queries to “account for the fact that sources respond at different speeds or may fail to respond at all.” Ex. 1001, 19:57-61, 24:13-17 (“In addition, the use of asynchronous queries that may result in rapid and graceful failure of some queries or commands may allow the system 90 to robustly return results quickly and in a manner that seems natural to the user.”).

210. As Ross explains, its external applications (*domain agents*) operate autonomously and thus send requests and/or commands *asynchronously* to local or external information sources apart from the main programming flow of recognizing and processing the speech input. Ex. 1005, 9:46-50 (“[T]he command is applied to the speech enabled application as needed and performing the function of the command.”), 10:19-23 (“[T]he command is applied to the selected speech enabled application as needed and performing the function of the command.”), Ross 12:6-13.

211. Moreover, Ross discloses that its method has “the capability to proceed with multiple execution requests simultaneously,” demonstrating asynchronous programming. Ex. 1005, 5:1-7.

212. As discussed for limitation 5[g], the external applications (*by the selected at least one or more domain agents*) create a “translated” query based on the at least one natural language query or command (*one or more queries based on the at least one query and/or command*). See Section X.C.8 (Ground 1, 5[g]).

213. A person of ordinary skill in the art would have further understood that by performing the function of the command the external applications (*domain agents*) send the command or query to the target application (*a local or external information source to execute the request*). Ex. 1005, 5:1-7.

214. Ross explains that after the utterance is translated into the appropriate commands or queries, they are then “dispatched to the target application 26.” Ex. 1005, 5:55-56. For example, for electronic mail application 26, commands may include “Open the first message” or “Create a message.” Ex. 1005, 8:1-6. Similarly, for calendar application 26, commands may include “Open the first appointment” and “Create an appointment.” Ex. 1005, 8:19-24. The external applications (*domain agents*) would send these commands to the electronic mail or calendar sources (*local or external information sources*) to execute.

D. Dependent Claims 6-7

1. **Claim 6:** “*The method of claim 5, further comprising generating, by the parser, the at least one query and/or command based on the natural language utterance.*”

215. In my opinion, Ross discloses this limitation.

216. Ross discloses that the context manager generates (*generating, by the parser*) the translated utterance (*at least one query and/or command*) *based on the spoken language utterance (natural language utterance)*. Ex. 1005, 7:29-33 (“In step 110, the context manager 50 sends the translated utterance 74 to the speech enabled application 26 indicated by the selected context 72 to perform the action indicated by the translated utterance 74.”), Figs. 4-5.

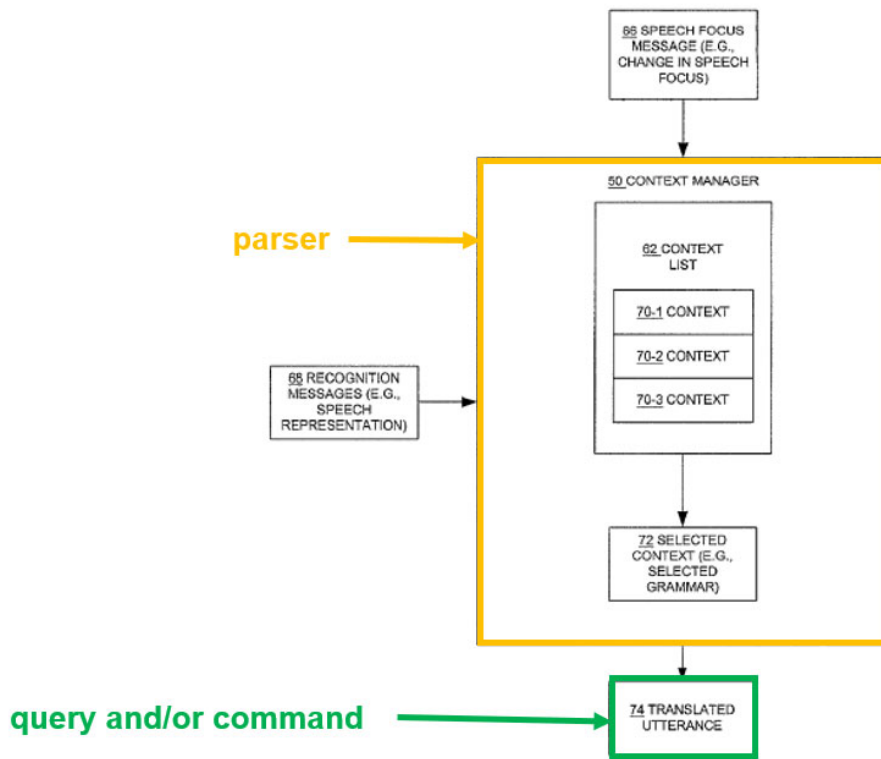


FIG. 4

Ex. 1005, Fig. 4 (annotated)

2. **Claim 7:** “The method of claim 6, wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.”

217. In my opinion, Ross discloses this limitation.

218. As discussed for claim 6, Ross discloses that the context manager generates (*generating, by the parser*) the translated utterance based on the spoken language utterance (*at least one query and/or command based on the natural language utterance*). See Section X.D.1 (Ground 1, Claim 6). Ross further discloses

that the translated utterance (*at least one query and/or command based on the natural language utterance*) can include commands to multiple applications (*comprises a plurality of queries and/or commands*).

219. Ross explains that the speech center “provides the capability to proceed with multiple execution requests simultaneously, to queue up additional script commands for busy applications 26, and to track the progress of the execution, informing the client when execution of a script is in progress or has completed.” Ex. 1005, 5:2-7. Accordingly, “[t]he spoken utterance [can] include[e] a command that is able to be associated with multiple applications.” Ex. 1005, cl. 1. This allows the system to avoid “the time, expense, and footprint of duplicating speech services in multiple applications.” Ex. 1005, 2:14-16.

XI. OPINIONS ON GROUND 2

220. In my opinion, claims 5-8 are rendered obvious by Ross (Ex. 1005) and Ittycheriah (Ex. 1007) in further view of Glover (Ex. 1011).

A. A POSITA Would Have Been Motivated to Combine

221. In my opinion, a person of ordinary skill in the art would have been motivated to implement Glover’s meta-search functionality as an external application that can interface with Ross’s system so that the system can gather relevant information from the Internet in response to a user utterance requesting a search.

222. Ross discloses that its external applications “perform the function of the command” in the natural language utterance. Ex. 1005, 9:44-50, 12:6-13. Ross does not detail how all its external applications obtain and process information to perform the function of the command. For example, Ross does not detail how information is sourced from the Internet. Thus, a person of ordinary skill in the art would have been motivated to look for and rely on an Internet searching mechanism to implement retrieving and processing responses to a user’s search queries (such as those disclosed in Glover). This is particularly true given the proliferation of the Internet at the time, and making it a standard feature for any multi-application system.

223. A person of ordinary skill in the art would have looked to implement a metasearch engine, like that in Glover, as one of the external applications of Ross. Glover includes a dispatcher that operates “similar to the query processor of a conventional search engine,” as it “generate[s] search engine requests from the user’s input,” to multiple search engines. Ex. 1011, ¶38. A person of ordinary skill in the art would have found a metasearch engine, like Glover, desirable because it interfaces with the Ross system in a way that would have allowed for a broad search over several Internet databases to provide “improve[d] coverage compared to a single search engine.” Ex. 1011, ¶35.

224. A person of ordinary skill in the art would be particularly motivated to implement Glover’s metasearch functionality because it uses “selective retrieval,” which “predicts the relevance of documents returned by regular search engines.” Ex. 1011, ¶5. In particular, Glover discloses a metasearch engine that performs continuous searching based on whether the returned results exceed a pre-determined confidence threshold. Ex. 1011, ¶83. For those documents below the threshold, Glover’s metasearch engine performs additional searches until it is able to have confidence that the returned result provides relevant results. Ex. 1011, ¶83. This “computation of the confidence of the relevance prediction,” can be done using “similarity measures..., or machine learning methods such as neural networks or support vector machines.” Ex. 1011, ¶¶85-86.

225. By adopting Glover’s “selective retrieval,” Ross’s ability to search the Internet and return results would benefit from “improve[d] execution time, resource usage, and throughput by requiring fewer retrieval requests compared to content-based metasearch engines.” Ex. 1011, ¶5. This would advance Ross’s objective of “avoid[ing] the time, expense, and footprint of duplicating speech services in multiple applications.” Ex. 1005, 2:14-16.

226. Accordingly, a person of ordinary skill in that art would have found that Glover is particularly well-suited to include in a speech-based system because it automates a lot of steps that a user would otherwise have to perform manually in

refining a search to obtain good results. Those search refinements would otherwise be more difficult to achieve with a speech interface, so a POSITA would find that adding Glover's metasearch engine to Ross would be a good improvement.

B. A POSITA Would have Had a Reasonable Expectation of Success

227. In my opinion, a person of ordinary skill in the art would have been able to implement a metasearch engine as an external application that could be invoked by Ross's speech center to invoke Internet searching. Further, there is nothing in Ittycheriah that would prevent the combination.

228. Glover's metasearch engine is a natural extension of Ross's ability to process natural language commands and questions by expanding the sources of information it can search to information on the Internet.

229. As Ross provides, its external applications can communicate with Ross's conversation manager and speech center through the "external application interface." Ex. 1005, 5:8-21. Because Ross already provides an interface for the external applications (*domain agents*) to communicate with Ross's conversation manager, it would be a simple adjustment to implement a network interface for the dispatcher of Glover to create queries and commands to obtain information from the Internet to respond to a user's utterance.

230. In my opinion, a person of ordinary skill in the art would have had the skill to implement and the reasonable expectation of success in achieving such a

combination because it would have involved a combination of known technologies (e.g., known systems for speech recognition and Internet searching based on user queries) according to known methods (e.g., known methods of speech recognition engines that convert speech into text strings and using text strings representing user's queries to perform Internet searching) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke various speech-enabled applications to retrieve information from the Internet.

C. Independent Claim 5

231. In my opinion, Ross and Ittycheriah in further view of Glover discloses independent claim 5.

1. Preamble through Limitation 5[f]

232. In my opinion, Ross in view of Ittycheriah renders obvious the preamble and limitations 5[a]-5[f] for the reasons discussed in Section X.C.1-7 (Ground 1, Claim 5[preamble]-5[f]).

2. Limitation 5[g]: “creating, by the selected at least one of the plurality of domain agents, one or more queries based on the at least one query and/or command;”

233. To the extent that Ross alone does not disclose limitation 5[g], Ross in view of Glover renders obvious this limitation.

234. Glover discloses that its metasearch engine includes a dispatcher that “generate[s] database queries based on the input from the user interface.” Ex. 1011,

¶38. Specifically, Glover’s dispatcher includes a “query generator 32,” that “modif[ies] queries appropriately for each source.” *Id.*, ¶39.

235. In the combination, Ross’s conversation manager would interface with Glover’s metasearch engine, including the dispatcher, to create “database queries based on input from the user interface.” Section XI.A (Ground 2, Motivation to Combine); Ex. 1011, ¶38. By implementing the ability of Ross’s conversation manger to interface with Glover’s metasearch engine, Glover’s dispatcher (*domain agent*) can generate database queries (*creating...one or more queries*) using the Ross’s translated utterance for locating information over the Internet (*based on the user’s query and/or command*).

3. Limitation 5[h]: “sending, by the selected at least one of the plurality of domain agents, the one or more queries in an asynchronous manner to one or more local or external information sources.”

236. To the extent that Ross alone does not disclose limitation 5[h], Ross in view of Glover renders obvious this limitation.

237. As discussed for limitation 5[g], Glover discloses that its dispatcher creates queries. *See* Section XI.C.2 (Ground 2, Limitation 5[g]). The dispatcher “provid[es]” the created queries to “a request generator 33 and then to a request submitter 34 for transmission to the World Wide Web,” which would include one or more search engines (*one or more local or external information sources*). Ex. 1011, ¶39; *see id.*, ¶35 (“A metasearch engine takes user queries and submits them to

multiple underlying search engines, and combines the results into a single interface.”). This is shown below in a diagram of a dispatcher:

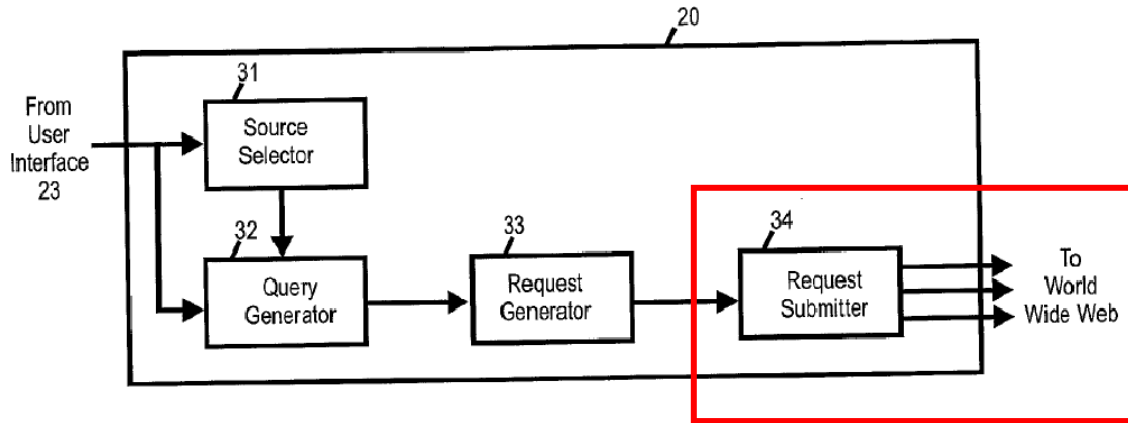


FIG. 3

Ex. 1011, Fig. 3 (annotated)

238. In my opinion, a person of ordinary skill in the art would have further understood that a query submitted to “multiple underlying search engines” is sending a query to *one or more local or external information sources*. The search engines search various disparate sources on the Internet, establishing them as different *information sources*.

239. In my opinion, a person of ordinary skill in the art would have also understood that the queries to the World Wide Web are performed *in an asynchronous manner*, as Glover states that its “steps may be performed in parallel.” Ex. 1011, ¶82. “For example, step 53 may be estimating the relevance of one or more results while step 52 is still sending queries 51 to and retrieving results 52 from

search engines.” *Id.* Since the steps are occurring in parallel, this indicates that each query is executed and operated independent of each other.

D. Dependent Claims 6-8

240. In my opinion, Ross and Ittycheriah in further view of Glover renders obvious dependent claims 6-8.

1. **Claim 6:** “*The method of claim 6, wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.*”

241. In my opinion, Ross discloses this limitation for the reasons discussed in Section X.D.1 (Ground 1, Claim 6).

2. **Claim 7:** “*The method of claim 6, wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.*”

242. In my opinion, Ross discloses this limitation for the reasons discussed in Section X.D.2 (Ground 1, Claim 7).

3. **Claim 8:** “*The method of claim 5, wherein one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries and send the duplicate queries in an asynchronous manner to different information sources.*”

243. In my opinion, Ross in view of Glover renders obvious this limitation.

244. As discussed for limitations 5[g] and 5[h], in the combination, the Ross conversation manager mediates between the external applications (*domain agents*)

including Glover’s metasearch engine such that the external applications create and send queries to different information sources based on the user’s input. Specifically, Glover’s dispatcher (*domain agent*) will send queries to various search engines (*different information sources*).

245. Glover discloses that as a general matter, “a metasearch engine takes user queries and submits them to multiple underlying search engines.” Ex. 1011, ¶35. Glover further discloses that its dispatcher can “choose search engines to query and...modify queries appropriately for each source.” *Id.*, ¶39. A POSITA would have understood that taking a single user query and submitting to multiple search engines discloses *creating duplicate queries* and *sending the duplicate queries* to *different information sources*, as a query must be duplicated in order to be sent to multiple sources. Fig. 3 (showing “request submitter” sending duplicate queries to the World Wide Web).

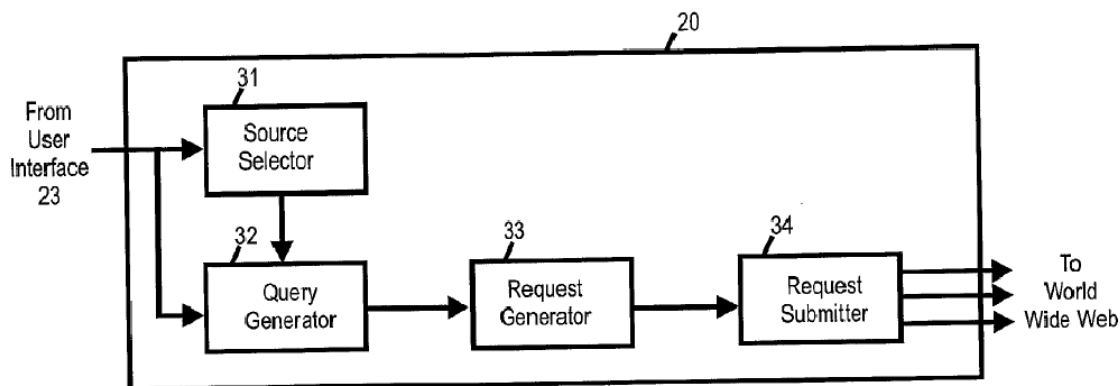


FIG. 3

246. For the same reasons as stated for limitation 5[h], a POSITA would also understand that Glover discloses sending the duplicate queries *in an asynchronous manner*. Section XI.C.3 (Ground 2, limitation 5[h]); Ex. 1011, ¶82.

XII. OPINIONS ON GROUND 3

247. In my opinion, claims 5-8 are rendered obvious by Hartono (Ex. 1006) and Baker (Ex. 1008).

A. A POSITA Would Have Been Motivated to Combine

248. A person of ordinary skill in the art would have been motivated to implement Baker's speech recognition method as Hartono's "automatic speech recognition system." Hartono suggests using any suitable off the shelf speech recognition system. At the time, Dragon System was well-known for developing speech recognition systems, and a person of ordinary skill in the art would have naturally looked at speech recognition systems available from Dragon Systems, like that taught in Baker.

249. Hartono discloses that speech is input into an "automatic speech recognition" module, shown below labeled as item 102a in Figure 2:

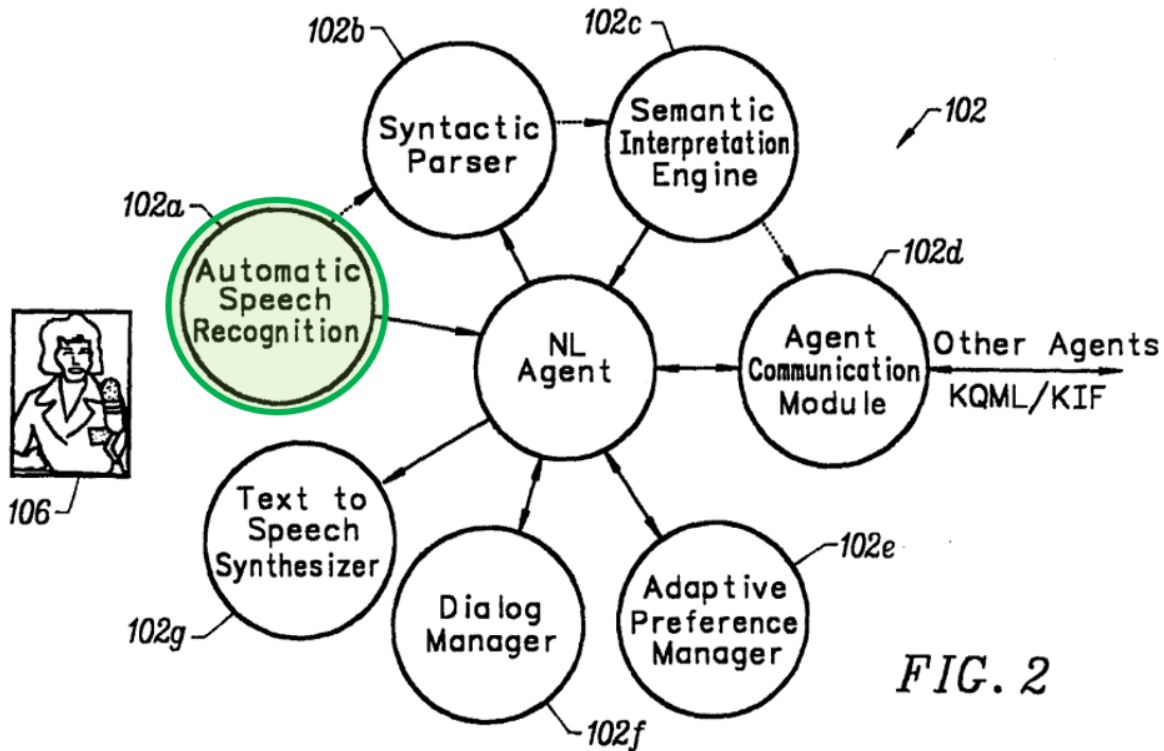


Figure 2, Ex. 1006

250. Hartono recognizes that by 1999, “automatic speech recognition systems for speech input are readily and commercially available off the shelf.” Ex. 1006, 11:4-12. Hartono thus discloses that “[a]ny suitable off the shelf speech recognition systems may be used as the automatic speech recognition system 102a in the natural language interface system 100 of the present invention.” Ex. 1006, 11:4-12.

251. Consequently, a person of ordinary skill in the art reading Hartono would have been motivated to look for commercially available speech recognition systems to use to perform the method steps described in Hartono.

252. Baker is such a speech recognition system developed by Dragon Systems, Inc. Ex. 1008, Cover at [73] (identifying assignee as “Dragon Systems, Inc.”). Dragon Systems was a well-known developer of speech recognition software in the mid-to-late 1990s, developing speech recognition software such as NaturallySpeaking. Ex. 1005, 1:16-21; Ex. 1023 (Dragon Naturally Speaking 5 User’s Guide (2000)).

253. A person of ordinary skill in the art would have also been motivated to implement the speech recognition system of Baker as a way to quickly complete the speech recognition process based on both acoustic probabilities and context probabilities by providing a probability and rank signal that Hartono can then use to determine which context is most suitable.

254. In addition, a person of ordinary skill in the art would have been motivated to use Baker’s speech recognition system with Hartono’s because Baker recognizes words “on the fly” using existing and on-line reference sources. As Baker explains, “conventional training techniques and their associated language models are difficult to adapt or edit” because they require new training data before the statistical model can be helpful. Ex. 1008, 4:36-38. “For instance, each new word, such as ‘obviousness’, require sufficient training and text and analysis to develop a meaningful statistical model of the word.” *Id.*, 4:38-40.

255. The ability to dynamically compute semantic similarity allows Baker to process commands more quickly over those methods that require building data-intensive statistical language models or data-training sets. Ex. 1008, 3:46-4:56 (describing the drawbacks of prior art systems based on “precomputed statistical counts” and “vocabulary training.”), 4:57-61 (“there is a need in the art of word recognition for improved systems and methods that can recognize words in a data string as a function of context and meaning, and in particular, there is a need for systems and method that require less training data and less training time”), 6:18-22 (“the invention provides methods and systems for simply and efficiently generating language knowledge, including generating lists of passages selected from one or more reference sources which can include a reference source that is accessed on-line.”), 13:66-14:13 (“The probability that a specific choice word is substantially similar to the current word being identified is increased if there is a match between either the choice word or one of its associated passage words and one of the context words in the data string.”).

256. A person of ordinary skill in the art would be motivated to use Baker’s faster processing in speech recognition to more quickly respond to specific user requests. Speed is particularly important when there is real-time interaction with the user, like the adaptive interaction in Hartono.

B. A POSITA Would have Had a Reasonable Expectation of Success

257. A person of ordinary skill in the art would have had a reasonable expectation of success in combining the teachings of Hartono and Baker.

258. A person of ordinary skill in the art would have understood that Baker's speech recognition system would be compatible with Hartono, as Baker discloses recognizing "words, or a series of words," for "translat[ing] into machine readable form." Ex. 1008, 1:5-15. Hartono expressly states that "details of the speech recognition methods and systems are not described," specifically teaching that "[a]ny suitable off the shelf speech recognition systems may be used" like those of Dragon Systems as described in Baker. Ex. 1006, 11:4-12.

259. Hartono also employs the Government and Binding approach to its syntactic parsing, which is based on the "implied syntactic structure in English language sentences." Ex. 1006, 12:1-10. Baker's speech recognition system that recognizes words based on their context within sentences, is thus compatible with Hartono's syntactic parser, which looks aims to accurately identify noun phrases, verb phrases, and adjective phrases. Ex. 1006, 12:15-13:10.

260. A person of ordinary skill in the art would have also had the skill to implement and the reasonable expectation of success in achieving such a combination because it would have involved a combination of known technologies (e.g., known systems for speech recognition and processing natural language

requests) according to known methods (e.g., known methods of speech recognition engines that convert speech into text strings that can be used by software applications) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke various speech-enabled applications on a computer.

C. Independent Claim 5

261. In my opinion, Hartono in view of Baker renders obvious independent claim 5.

1. Preamble: “A method for responding to a user generated natural language speech utterance, the method comprising:”

262. In my opinion, Hartono discloses the preamble to the extent it is limiting.

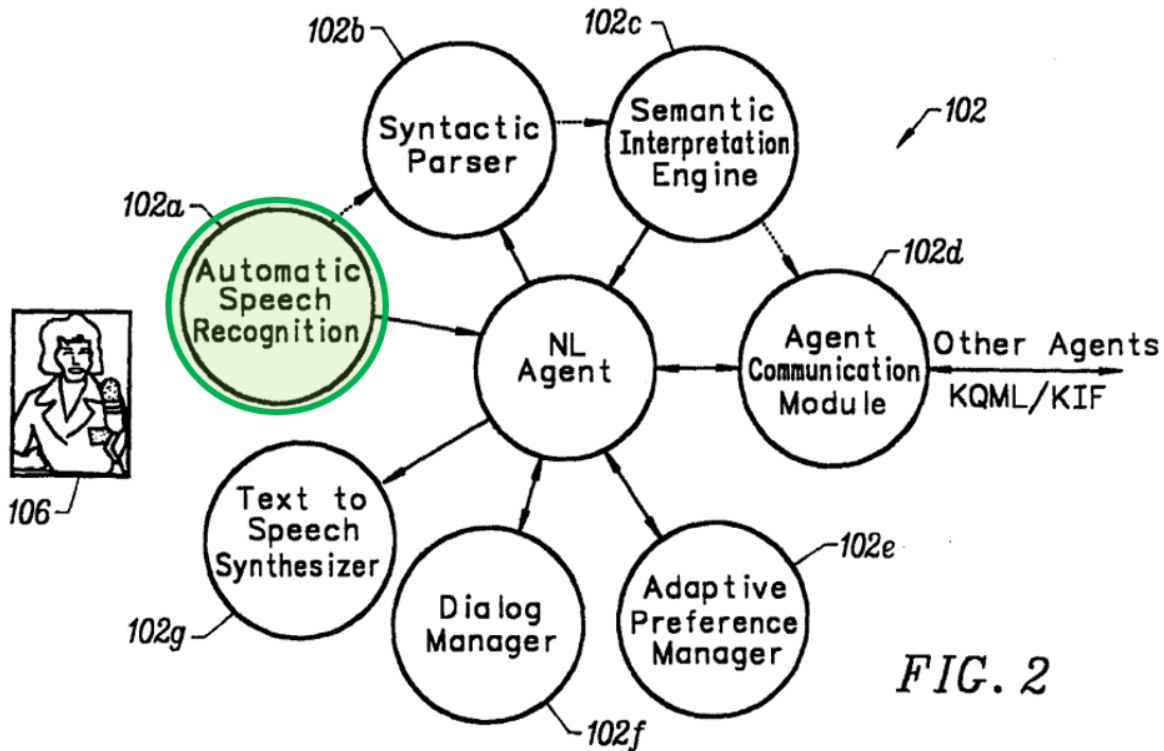
263. Hartono discloses outputting results in response to (*method for responding to*) receiving commands spoken by a user or other user input (*user generated language speech utterance*). In particular, Hartono discloses a “method for receiving commands and/or adaptively outputting results using a natural language speech interface.” Ex. 1006, 8:8-10; *see also* 1:19-23 (“provides a method for receiving commands, executing received commands and adaptively interacting with the user using a natural language interface.”).

264. As Hartono explains, “[i]n one embodiment, the natural language interface for a computer system includes a natural language agent adapted to receive and interpret the natural language user input and to output an output command and at least one application agent adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program.” Ex. 1006, 4:20-25; *see also* 5:16-24, 10:13-20.

2. Limitation 5[a]: “recognizing, by a speech recognition engine, one or more words in the user generated natural language speech utterance;”

265. In my opinion, Hartono in view of Baker renders obvious this limitation.

266. As I discussed in the Overview, Hartono discloses a system for processing natural language:



267. Hartono recognizes that “[a]utomatic speech recognition systems for speech input are readily and commercially available off the shelf. Any suitable off the shelf speech recognition systems may be used as the automatic speech recognition system **102a** in the natural language interface system **100** of the present invention.” Ex. 1006, 11:4-12.

268. Thus, a POSITA would have been motivated to implement an off-the-shelf speech recognition system, like that disclosed in Baker, in the system of Hartono because that is what Hartono teaches.

269. As discussed in Section VIII.D (Baker Overview), Baker's word recognition system is a *speech recognition engine* that *recognizes one or more words in the user generated natural language speech utterance*.

270. Baker teaches recognizing words in two ways. First, Baker describes systems and methods for “detect[ing] words within portions of speech signals and compar[ing] these detected words to acoustic models of known words, or sequence of acoustic models of known words.” Ex. 1008, 18:13-16.

271. Notably, Baker defines the term “word” to mean “a word or a phrase and may be spoken, written, or represented as electronic data signals.” Ex. 1008, 6:1-3.

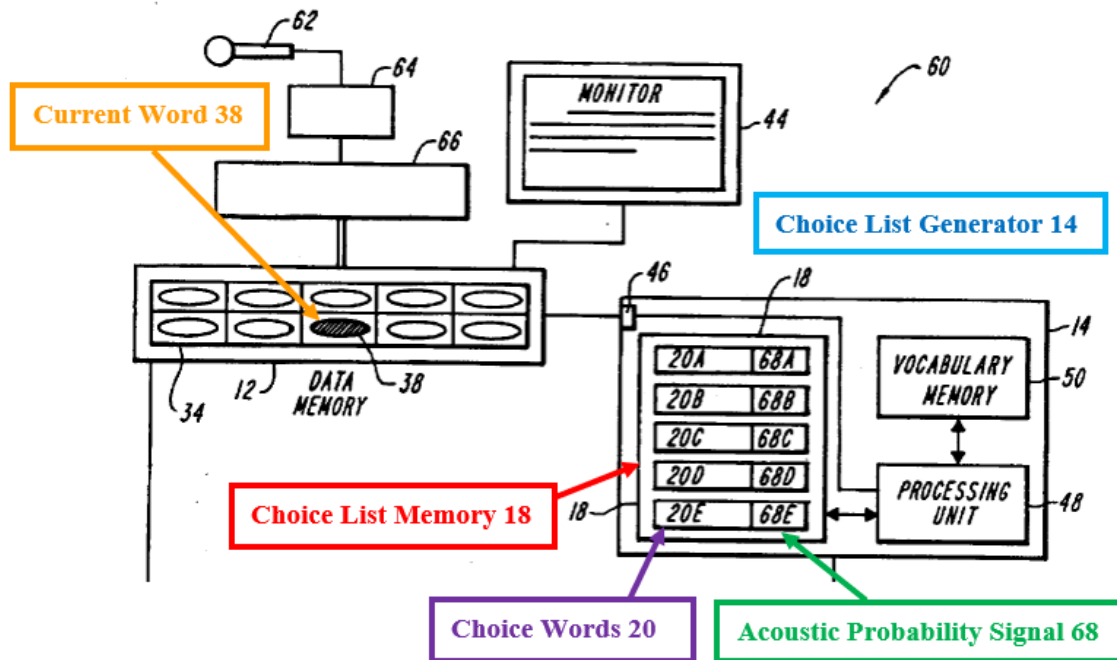
272. Accordingly, Baker's use of the term “word” also encompasses “phrases” since Baker expressly defines “word” to include phrases.

273. Baker explains that the word (or phrase) is associated with a probability signal. “This comparison leads to a probability signal that indicates the certainty that the detected word is substantially acoustically similar to the word associated with the acoustic model.” Ex. 1008, 18:16-19.

274. Words with an acoustic probability signal indicating a “high degree of certainty” are deemed recognized (referred to as “context words”). Ex. 1008, 18:16-19.

275. However, words with a “low degree of certainty” indicate that the word has not been confidently identified (referred to as the “current word”). Ex. 1008, 18:22-24.

276. Second, for those “unrecognized words,” Baker creates a “choice list 18” that “store[s] the words 20 associated with the acoustic models from the vocabulary memory 50 which are acoustically similar to the current ambiguous word 38 being identified, as illustrated below in the following excerpt of Figure 2:



Ex. 1008, Fig. 2 (excerpted and annotated)

277. Baker explains that each “choice word 20A-E” has a corresponding “acoustic probability signals 68A-E” that represent “the degree of certainty that the

word 20 associated with the acoustic model is substantially representative of the current word 38 being recognized.” Ex. 1008, 18:30-35.

278. Baker further teaches that probability signals 68A-E “can represent the acoustic model probability adjusted by an n-gram probability, or any suitable language model.” Ex. 1008, 18:35-37, 20:30-46 (“The probability signals 68A-68E can represent the likelihood that the associated word choice 20A-20E is substantially representative of the current word 38 being recognized.”), 22:55-64.

279. Once a “choice word 20” has been identified, Baker teaches generating a “word list” of words that “have a semantic association with the choice word 20.” Ex. 1008, 14:57-69.

280. The “semantic association” for each choice word (which Baker also refers to as “passage words” (labeled as item 24 in Fig. 2)) in the “word list” is derived from a “reference database 40” which can be a “dictionary, thesaurus, antonym dictionary, medical dictionary, legal dictionary, scientific and engineering dictionary, or other type of reference source.” Ex. 1008, 20:55-58 (“The word list generator 16 can access via a transmission path the reference database 40 and can identify information in the reference database 40 that relate to the topic, theme, or subject of a choice word 20.”).

281. Baker next “perform[s] a matching function to calculate the number of matched words” between those words recognized by the speech analysis unit 66 with

high confidence (the “context words”) with the words in the “word list” (the “passage words”). Ex. 1008, 21:19-47.

282. By looking at the “relative frequency of matches between the “context words” and the “passage words,” the data processor calculates a “probability signal 28 representative of the likelihood that the choice word or phrase 20 substantially represents the utterance currently being recognized.” Ex. 1008, 21:19-47, 21:49-22:6, Figs. 2-4.

283. Baker’s system generates a “rank signal 30” that “represents the likelihood...that a choice word **20** is substantially similar to” the word being recognized” as a function of “context matching and acoustic model matching” probabilities. Ex. 1008, 23:22-27.

284. A table correlating each “choice word 20” (words selected based on acoustic-matching) with context-matching “probability signal 28” and “rank signal 30” is shown below in an annotated excerpt of Figure 2:

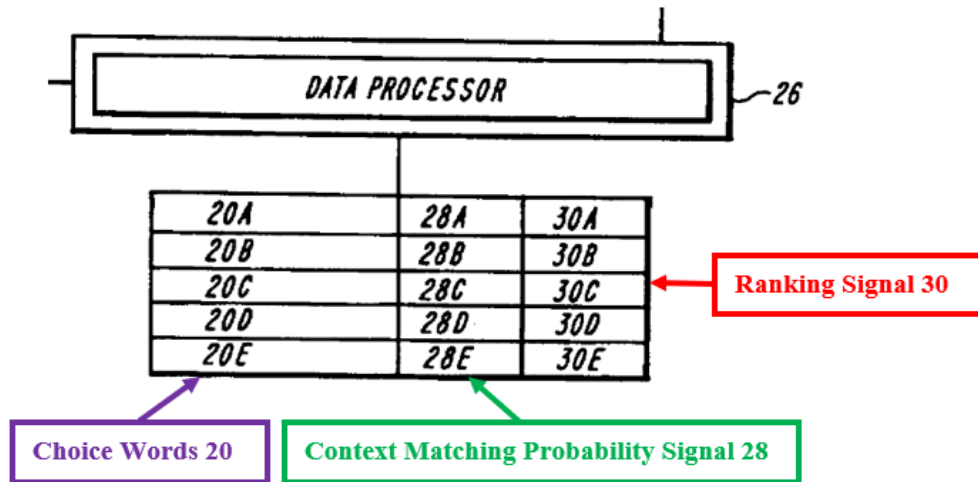


Fig. 2 (excerpted and annotated)

See also Ex. 1008, 21:48-22:6 (describing computing rank signal 30 “given two constraints, the probability signal 28 and the probability signal 68” according to techniques “well known in the field of mathematics and any of the techniques capable of generating a probability signal given a set of constraints”)

285. By looking at both acoustic matching and context matching, Baker is thus able to *recognize one or more words in the user generated natural language speech utterance.*

3. Limitation 5[b]: “receiving, at a parser, keyword and associated prior probabilities or fuzzy possibilities from a system agent or an active domain agent of a plurality of autonomous executable domain agents;”

286. In my opinion, Hartono in view of Baker renders obvious this limitation.

287. Baker's speech recognition system is a *system agent* because it is a component of Hartono's system that provides the basic service of recognizing words in the utterance and outputs words or phrases along with probability and rank signals. *See also* Section X.C.3 (Ground 1, Limitation 5[b]) (discussing the '825 patent's description of *system agent*).

288. In the combination, Baker's output is the recognized words or series of words of a spoken utterance "just as if the data were being typed." Ex. 1008, 1:4-7, 1:32-36.

289. Those recognized words or series of words (*keyword*) along with a probability and rank signal (*associated prior probabilities or fuzzy possibilities*) are then inputted into Hartono's natural language "syntactic parser" and "semantic interpreter." Ex. 1006, 11:14-16:11, Fig. 2; *see also* Section XII.C.4 (Ground 3, 5[c]).

290. Hartono receives the recognized speech from Baker and its associated prior probabilities or fuzzy possibilities at its "natural language syntactic parser 102b" and "natural language semantic interpreter 102c" (*parser*). *See* Section XII.C.4 (Ground 3, 5[c]).

291. Thus, in the combination, Hartono's syntactic parser and semantic interpreter receives (*receiving, at a parser*) recognized words (*keyword*) and

probability and rank signals (*associated prior probabilities or fuzzy possibilities*) from Baker's speech recognition system (*from a system agent*).

4. Limitation 5[c]: “determining, for the natural language speech utterance, a score for each of at least two possible contexts, wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities;”

292. In my opinion, Hartono in view of Baker renders obvious this limitation.

293. As I discussed for limitation 5[b], Hartono's “natural language syntactic parser 102b” and “natural language semantic interpreter 102c” (*parser*) receive the recognized words (*received keywords*) and probability and rank signals (*associated prior probabilities or fuzzy possibilities*) from speech recognition engine (*system agent*) of Baker. See Section XII.C.3 (Ground 3, Limitation 5[b]).

294. Upon receiving the recognized words, Hartono then *determin[es]...a score for each of at least two possible contexts* for the *natural language speech utterance* by parsing through the recognized words to determine the best fit for the syntactic parse.

295. Hartono discloses syntactically parsing the recognized word into a structure using a Government and Binding approach (a computation method based on linguistic theory). Ex. 1006, 12:1-20.

296. This parsing “allow[s] the computer to map the verb into a computer command action, with the noun phrase (NP) as the object and the adjective phrase (AP) as properties of the object.” Ex. 1006, 13:8-10.

297. Hartono’s semantic interpreter then applies “context sensitive methodologies” to the syntactic parse to *determine a score for each of at least two possible contexts*. Ex. 1006, 13:13-21.

298. Specifically, Hartono’s semantic interpreter uses a “reverse grammar generation mechanism” that maintains “a list or vector for each word and corresponding probabilities for each word in the list.” Ex. 1006, 14:10-20. This “associated vector or list,” for example, includes the words “‘I,’ ‘eye,’ or ‘aye,’” and has “corresponding probabilities of 80%, 15%, and 5%”, resulting in different permutations of the natural language utterance, scored from highest to lowest probability. Ex. 1006, 14:10-20. Consequently, the *possible contexts* are “I”, “eye, or “aye.”

299. Hartono uses these different probability-based permutations to “determine[] which words best fit the grammar of the syntactic parse,” which are *determined scores for possible contexts*. Ex. 1006, 15:1-12.

300. A person of ordinary skill in the art would have understood that to determine “best fit,” the system would need to assign a value or score to the evaluated grammar of the syntactic parse to determine whether or not the word is

“suitable given the context.” To dig further into the case of the example presented by Hartono, we know that Hartono proceeds through the permutations of the syntactic parse from the word with the highest probability to the word with the lowest probability (*associated prior probabilities or fuzzy possibilities*), checking to see which alternatives best fit the grammar of the syntactic parse. It does not simply take the permutation with the highest probability – it must also fit the grammar.

301. In the context of an agent for dictating email messages using speech, the phrase “I [0.80] want to go home” may very well be represented in the grammar. However, the phrases “Eye [0.15] want to go home” and “Aye [0.05] want to go home” will not be in the grammar. Here we have also indicated the probability of the word in question within each syntactic parse.

302. Hartono’s system starts by testing “I [0.80] want to go home” because it has the highest probability and determines that in fact fits the grammar. In other words, the word “I” is suitable given the context, which is an email dictation agent. Neither of the other two permutations needs to be considered because neither of them makes much sense when dictating an email.

303. However, in a more complex case where there are multiple plausible permutations of the syntactic parse, we see more instructive behavior. For example, consider an interaction between a ticket selling agent and a user when the prompt from the selling agent has been “Would you like a free concert ticket?”:

(A) “I would like two [0.60]” or

(B) “I would like to [0.40]”

In this case, there are two possible interpretations for the last word in the user’s response, and the syntactic parser has determined that “two” has a higher probability.

So the permutation “I would like two [0.60]” is tested first.

304. But we can envision that in the case of this particular ticket selling agent, a user is not allowed to request more free tickets than they have been offered; they are only allowed to accept or decline the offer of a single free ticket. In this context, “two” is not suitable even though it has the highest probability – it is not represented in the grammar, so the overall score is low even though the probability of the word is high. So the semantic interpretation engine then turns to the second alternative, “I would like to [0.40]” and finds that it is suitable given the context – the score is suitably high enough that the system takes this as the interpretation of the user’s utterance.

305. Also consider a situation where there can be two contexts, and the user makes the following utterance: “I need 8 ounces of honey.” Perhaps the user has access to two agents who can take instructions: one for ordering groceries to be delivered, and one for online banking. Say that the syntactic parser returns the following two options for the final word:

(A) “I need 8 ounces of money [0.60]” or

(B) “I need 8 ounces of honey [0.40]”

Because “I need 8 ounces of money [0.60]” has the highest probability, the semantic interpretation engine tries it first, checking with the grammar for the online grocery agent, where the permutation is not considered suitable because the grocery agent does not arrange to deliver money. When checking with the online banking agent, the permutation is also not suitable because the agent’s grammar does not include the concept of “ounces” of money.

306. With both the grocery agent and the banking agent returning low scores, the semantic interpretation engine turns to the permutation with the lower probability: “I need 8 ounces of honey [0.40]” When checking with the banking agent, its grammar returns a low score because “honey” does not correspond to the currency or other monetary instruments it delivers to customers. Finally, when checking with the grocery agent, its grammar returns a suitable score because it is programmed to interpret “8 ounces of honey” as part of a plausible grocery order.

307. The suitability given the context is based on the recognized word and probability and rank signals (*wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities*). As Hartono teaches, “[i]f the word is not suitable given the context, the next word having each next highest probability is then evaluated and determined if that word is suitable given the context, until a suitable word is determined.” Ex. 1006, 15:8-

12. Consequently, Hartono will iterate through the recognized words received from Baker depending on the probabilities and ranks provided.

308. Thus, in my opinion, Hartono in view of Baker renders obvious *determining, for the natural language speech utterance, the suitability (a score) for the grammar of the syntactic parse for the different permutations of the words (for each of at least two possible contexts) based on the recognized words and probability and rank signals received from Baker (wherein the scores are determined based on the received keyword and associated prior probabilities or fuzzy possibilities).*

5. Limitation 5[d]: “determining by the parser, a domain for the user generated natural language utterance based on the recognized one or more words of the natural language utterance and the determined scores for each of the at least two possible contexts;”

309. In my opinion, Hartono discloses this limitation.

310. Once the semantic interpreter has scored the words in the recognized natural language utterance, it then *determine[es]* an application-context for the utterance (*a domain for the user generated natural language utterance*) by “tak[ing] the syntactic parse of the spoken language request and map[ping] it to a generic concept frame,” such as that shown below in Table 1:

TABLE I		
ACTION-CONCEPT	TOPIC CONCEPT	APPLICATION-CONTEXT
Show	Email	Email application
Show	Email address	Address book application
Delete	Email	Email application
Show	General help	Natural language agent

Ex. 1006, Table I

See also Ex. 1006, 13:18-21, Table I

311. Specifically, the semantic interpreter uses verbs in the recognized words or phrases as “action-concepts” and nouns that “describ[e] the objects on which the actions are performed” as “topic-concepts.” Ex. 1006, 14:1-7.

312. Hartono “uses the combination of the action-concept and the topic-concepts” to “determine which task agent should handle the request.” Ex. 1006, 15:13-19.

313. The determined *domains* are shown in Table I as the “Application-Context,” as “email application” or “address book application.” Ex. 1006, Table I, 6:2-5 (“The natural language interface parses the user’s input and semantically maps it into a knowledge concept structure. The system then determines which application context should be responsible for interpreting and executing that command concept.”).

6. **Limitation 5[e]: “selecting at least one of the plurality of autonomous executable domain agents based, at least in part, on the determined domain, wherein each of the plurality of domain agents is configured to respond to queries and/or commands within a particular domain, wherein the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands;”**

314. In my opinion, Hartono discloses the limitation.

315. As I discussed for Ground 1, a person of ordinary skill in the art would have understood a *domain agent* as claimed to be an autonomous executable software module specific to each application area. *See* Section X.C.6 (Ground 1, Limitation 5[e]).

316. The '825 patent uses an agent-based architecture that was well-known at the time. Hartono also “utilizes an agent-based architecture” that employs “application class task agents for each class of applications,” as illustrated below in Figure 1:

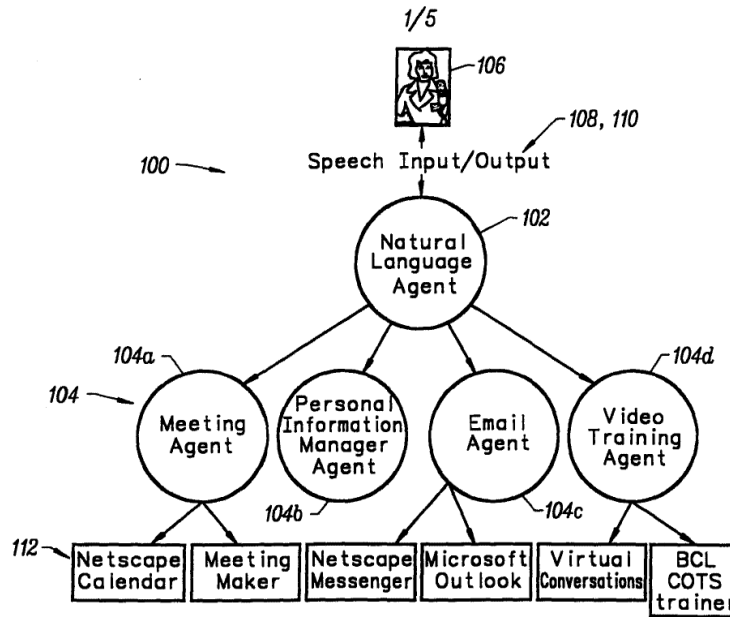


FIG. 1

Ex. 1006, Fig 1,

See also Ex. 1006, 8:10-13, 9:9-14 (“Examples of the task agents 104a-d shown in FIG. 1 are meeting agent 104a, personal information manager agent 104b, email agent 104c, and voice training agent 104d. Each task agent 104a-d outputs to the natural language agent 102 which then delivers the natural language output 110 to the user 106.”).

317. A person of ordinary skill in the art would have understood that Hartono’s “task agents” (also referred in Hartono as “application class agents”) correspond to the claimed *autonomous executable domain agent* in the ’825 patent because they are software modules specific to an application or application area (meetings, email, personal information manager, video training, etc.). Ex. 1006,

9:15-20 (“Each of the back-end application class agents 104a-d works with a class of one or more existing computer applications.”), 23:19-20 (“Each application class agent 104 preferably works with a single class of applications 112 that have similar conceptual operations.”), 28:7-29 (Table IV listing “task agents”).

318. A person of ordinary skill in the art would have also understood that each of Hartono’s task agents are *configured to respond to queries and/or commands within a particular domain, wherein the particular domain indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands*. As Hartono explains:

Each task agent for a class of applications is preferably provided with its own set of semantically correct sentences, semantic actions and semantic mapping. Each task agent thus serves as the common user interface for the corresponding class of applications under the assumption that each application within the class accomplishes the same or generally overlapping sets of tasks. Ex. 1006, 27:17-28:3.

319. To that end, Table IV of Hartono provides a listing of different task agents and sample tasks in their *particular domain*. Ex. 1006, Table IV.

320. For example, the “mail” task agent has expertise in responding to commands to send, receive, compose and view email. Ex. 1006, Table IV.

321. By way of another example, the “address” task agent has expertise in managing address books. Ex. 1006, Table IV.

322. Also as an example, the “calendar” task agent has expertise in managing calendar, setting meetings and setting ticklers. Ex. 1006, Table IV. *See also* Ex. 1006, 29:2-30:6.

323. Accordingly, based on their domain of expertise, Hartono’s task agents each “include[] a process-specific execution module for each application.” Ex. 1006, 30:15-20 (“For example, the word processing task agent may include an execution module for MICROSOFT WORD”), 36:7-9 (“application agent...[is] adapted to output an executable instruction to an application program”).

324. Hartono further discloses using a “task interpretation engine” to *select [] at least one...domain agent[] based, at least in part, on the determined domain* because the task interpretation engine selects the domain agent based on the action-concept and topic-concept for that identified application-context. Ex. 1006, 24:21-25:3 (“The task interpretation engine 104B receives the semantic frame representation as input. Based on the frame’s head verb (action request) and noun phrases (parameters), the task interpretation engine 104B invokes a routine that sends a set of requests to the task application wrapper 104A.”), 15:13-15 (“A combination of the action-concept and the topic-concepts are used to determine

which task agent should handle the request. If the request is for a specialized task agent, the request is routed to that specialized agent.”).

325. Thus, the task interpretation engine selects the task agent (*selecting at least one of the plurality of autonomous executable domain agents*) based on the application-context (*based, at least in part, on the determined domain*). Each task agent serves as a common interface for the class of applications in the application-context, with the ability to execute tasks, including queries and commands, understandable to that class of applications (*wherein the particular indicates an area of expertise within which the domain agent is capable of responding to the queries and/or commands*).

7. Limitation 5[f]: “providing at least one query and/or command based on the natural language utterance to the selected at least one of the plurality of domain agents;”

326. In my opinion, Hartono discloses this limitation.

327. Hartono discloses that its task agents are invoked by being *provid[ed] at least one query and/or command based on the natural language utterance*. Hartono explain that “[a]t least one application agent [is] adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program.” Ex. 1006, Abstract; *see also* 36:7-9 (Hartono’s “application agent” is “adapted to receive and further interpret the

output command from the natural language agent and adapted to output an executable instruction to an application program.”).

328. As shown in Figure 1 and described by Hartono, the natural language agent outputs the command based on the user speech input. Ex. 1006, 9:8-10 (“As in generally shown, the user 106 communicates an input phrase, command or sentence 108 to the natural language agent 102 which processes the input sentence and sends the input sentence to the appropriate one of the back-end application class task agents 104a-d.”), 4:20-25 (“the natural language interface for a computer system includes a natural language agent adapted to receive and interpret the natural language user input and to output an output command and at least one application adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program”), 5:4-15 (“Each application agent may include a semantic task interpreter adapted to generate the executable instruction from the output command of the natural language agent, and at least one application wrapper, each wrapper configured to communicate with a corresponding application program.”).

8. Limitation 5[g]: “creating, by the selected at least one of the plurality of domain agents, one or more queries based on the at least one query and/or command;”

329. In my opinion, Hartono discloses this limitation.

330. Hartono discloses that its task agents (*domain agents*) are invoked to output an executable instruction to an application program based on the received natural language command (i.e., *creating...one or more queries based on the at least one query and/or command*). Ex. 1006, Abstract (“[A]t least one application agent [is] adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program.”), 4:20-25, 5:4-15, 5:21-25 (“[O]utputting the output command to an application class agent, semantically interpreting the output command and generating an executable instruction from the output command, and outputting the executable instruction to an application program for execution by the application.”), 36:7-9 (Hartono’s “application agent” is “adapted to receive and further interpret the output command from the natural language agent and adapted to output an executable instruction to an application program.”).

9. Limitation 5[h]: “sending, by the selected at least one of the plurality of domain agents, the one or more queries in an asynchronous manner to one or more local or external information sources.”

331. In my opinion, Hartono discloses this limitation.

332. As I discussed for Ground 1, a POSITA would understand that the term “asynchronous” in the programming context refers to sending the queries to be executed autonomously, rather than being controlled by the main programming flow. *See* Section X.C.9 (Ground 1, Limitation 5[h]).

333. Hartono's application class agents (*domain agent*) *send the one or more queries in an asynchronous manner to one or more local or external information sources.*

334. First, Hartono discloses that its task agents (*domain agents*) operate autonomously (*see* limitation 5[e]) and thus send requests and/or commands *asynchronously* to local or external information sources apart from the main programming flow of recognizing and processing the speech input. *See* Ex. 1006, 30:15-20 ("For example, the word processing task agent may include an execution module for MICROSOFT WORD"), 36:7-9 ("application agent...[is] adapted to output an executable instruction to an application program").

335. Second, Hartono's application class agents "communicate[] with specific application 112 allowing the incorporation of existing applications into the architecture of the system," via content-specific wrappers. Ex. 1006, 24:4-13.

336. Accordingly, a POSITA would understand that Hartono's application task agents send the one or more queries to different types of "vendor-specific application programs" (*local or external information source*). Ex. 1006, 24:4-13. Moreover, Hartono discloses that these vendor-specific application programs would include web search engines over the Internet. Ex. 1006, Table IV (detailing a "web" task agent," with vendor-specific applications perform the tasks of "connect[ing], navigat[ing], search[ing] the internet/world-wide-web"), 26:1-6.

D. Dependent Claims 6-8

337. In my opinion, Hartono discloses dependent claims 6-8.

1. **Claim 6:** *“The method of claim 5, further comprising generating, by the parser, the at least one query and/or command based on the natural language utterance.”*

338. In my opinion, Hartono discloses the limitation.

339. Hartono discloses that the natural language syntactic parser and semantic interpreter generate (*generating, by the parser*) at least one operation based on the natural language input from the user (*at least one query and/or command based on the natural language utterance*). See Ex. 1006, 4:20-25 (“the natural language interface for a computer system includes a natural language agent adapted to receive and interpret the natural language user input and to output an output command and at least one application adapted to receive and further interpret the output command from the natural language agent and to output an executable instruction to an application program”), 5:4-15 (“Each application agent may include a semantic task interpreter adapted to generate the executable instruction from the output command of the natural language agent, and at least one application wrapper, each wrapper configured to communicate with a corresponding application program.”).

340. Hartono explains that “[t]he front end natural language 102 receives all natural language input and determines which of the available task agents 104 to pass

the natural language input interpreted by the front end natural language agent 102.”

Ex. 1006, 10:14-17. As explained for limitations 5[d] and 5[e], the syntactic parser and semantic interpreter determine and select which task agent (*domain agent*) to respond to the query. *See* Sections XII.C.5-6 (Ground 3, Limitations 5[d]-5[e]). As part of that process the syntactic parser and semantic interpreter also generates the operations. *See* Ex. 1006, 13:18-21 (“The natural language semantic interpreter 102c takes the syntactic parse of a spoken language request and maps it into a generic concept frame used to invoke the appropriate application method.”), Table I.

- 2. Claim 7:** *“The method of claim 6, wherein the at least one query and/or command based on the natural language utterance comprises a plurality of queries and/or commands.”*

341. In my opinion, Hartono discloses the limitation.

342. As discussed for claim 6, Hartono discloses that the natural language syntactic parser and semantic interpreter generate (*generating, by the parser*) at least one operation based on the natural language input from the user (*at least one query and/or command based on the natural language utterance*). *See* Section XII.D.1. (Ground 3, Claim 6). Hartono further discloses that the operation can include a set of requests for multiple different applications in order to perform the function of the command.

343. Hartono explains that “[t]he communication between the application class agent 104 and each different type of vendor-specific applications programs 112

is via an application wrapper 104A that translates the conceptual action to *a set of application specific operations.*” Ex. 1006, 24:4-7; *see also* 23:20-23 (“For example, different email applications generally perform the same conceptual actions of sending and receiving mail but performs these actions through different sets of steps.”).

3. **Claim 8:** “*The method of claim 5, wherein one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries and send the duplicate queries in an asynchronous manner to different information sources.*”

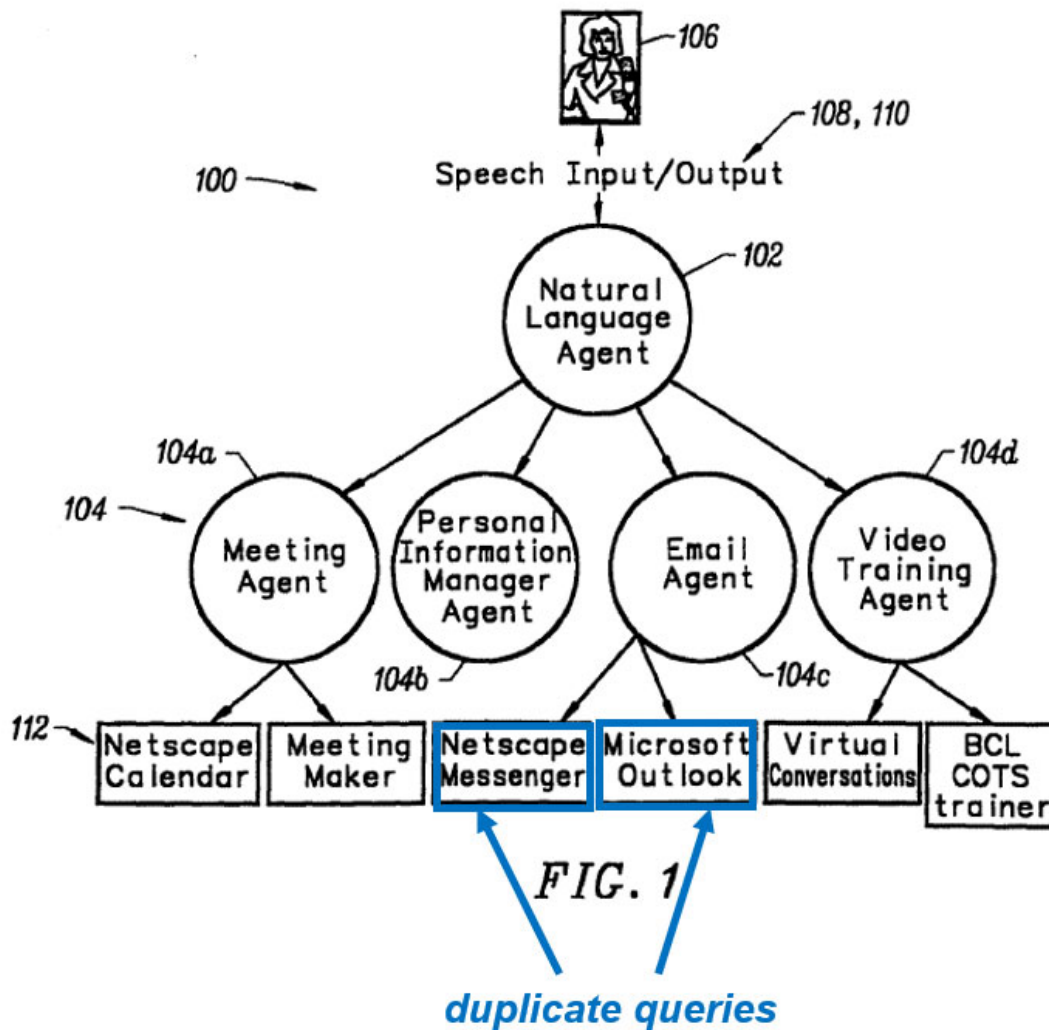
344. In my opinion, Hartono discloses this limitation.

345. Like the ’825 patent, Hartono discloses sending duplicate queries to different applications that perform the same task. *See* Section XI.D.3 (Ground 2, Claim 8) (discussing the ’825 patent’s description of *duplicate queries*).

346. Hartono discloses that the task agents are configured to translate an action to a set of application specific operations (*one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries*).

347. As discussed for limitation 5[e], Hartono’s “task agents” (also referred in Hartono as “application class agents”) correspond to the claimed *domain agents*. *See* Section XII.C.6 (Ground 3, Limitation 5[e]). Hartono explains that “[t]he communication between the application class agent 104 and each different type of

vendor-specific applications programs 112 is via an application wrapper 104A that translates the conceptual action to *a set of application specific operations.*” Ex. 1006, 24:4-7. Hartono further discloses that these operations are *duplicates* because each operation executes the same query for different applications. *See* Ex. 1006, 23:19-23 (“Each application class agent 104 preferably works with a single class of applications 112 that have similar conceptual operations”), Fig. 1 (below). “For example, an e-mail agent would have a wrapper for interacting with each email system such as NETSCAPE and MICROSOFT EXCHANGE.” Ex. 1006, 24:11-13; *see also* 26:25-3 (“An application wrapper interfaces with the particular address book application. The routines will handle the tasks as described above and will interface to the address book module for, e.g., MICROSOFT EXCHANGE, and NETSCAPE”).



Ex. 1006, Fig. 1 (annotated)

348. Hartono further discloses that the tasks agents (*one or more of the selected at least one of the plurality of domain agents*) will send the set of application specific operations (*send the duplicate queries*) to different particular applications, such as MICROSOFT EXCHANGE and NETSCAPE, independent of each other (*in an asynchronous manner to different information sources*).

349. Hartono explains that “the application class agent 104 communicates with specific applications 112 allowing the incorporation of existing applications into the architecture of the system 100.” Ex. 1006, 24:9-11. As discussed for limitations 5[e] and 5[h], the execution of these operations are *asynchronous* because they operate autonomously. See Sections XII.C.6 and XII.C.9 (Ground 3, Limitations 5[e] and 5[h]); *see also* 23:20-23 (“For example, different email application generally perform the same conceptual actions of sending and receiving mail but performs these actions through different sets of steps.”).

XIII. OPINIONS ON GROUND 4

350. In my opinion, claim 8 is rendered obvious by Hartono (Ex. 1006) and Baker (Ex. 1008) in further view of Glover (Ex. 1011).

A. A POSITA Would Have Been Motivated to Combine

351. A person of ordinary skill in the art would have been motivated to implement Glover’s metasearch functionality as a task agent to Hartono to aid in gathering relevant information from the Internet.

352. As discussed for limitation 5[h], Hartono discloses that its application task agents include agents with applications that perform Internet searches such as the “web” task agent, “Internet Product-Information lookup,” and “Internet Event look-up” agents. See Section XII.C.9 (Ground 3, Limitation 5[h]); Ex. 1006, Table IV, 39:1-13, 41:1-13.

353. Hartono thus already contemplates that its application task agents (*domain agents*) would be creating and sending queries over the Internet via its vendor specific applications.

354. Hartono does not detail how its task agents create and send queries over the Internet, and therefore, a person of ordinary skill in the art would have looked at well-known ways to implement such functionality, particularly those that would improve the system and align with Hartono's goals.

355. Accordingly, a person of ordinary skill in the art would have been motivated to implement Hartono in a way to query multiple information sources over the Internet to obtain the most relevant results (such as those disclosed in Glover).

356. This is especially true because Hartono's application task agents like the "Internet Product-Information lookup" and "Internet Event look-up agents" that look to find specific information (e.g., about events, products, and prices), in which it would be desirable to obtain high-quality, relevant information.

357. A person of ordinary skill in the art would look to implement in Hartono a metasearch engine like Glover that includes a dispatcher as a task agent that operates "similar to the query processor of a conventional search engine," as it "generate[s] search engine requests from the user's input," to multiple search engines. Ex. 1011, ¶38.

358. A person of ordinary skill in the art would have found a metasearch engine, like Glover, that acts as Hartono’s Internet-focused application task agents would have allowed for a broad search over several Internet databases to provide “improve[d] coverage compared to a single search engine.” Ex. 1011, ¶35.

359. A person of ordinary skill in the art would have been further motivated to implement Glover’s metasearch functionality because it uses “selective retrieval,” which “predicts the relevance of documents returned by regular search engines.” Ex. 1011, ¶5. Specifically, Glover discloses a metasearch engine that performs continuous searching based on whether the returned results exceed a pre-determined confidence threshold. Ex. 1011, ¶83. For those results below the threshold, Glover’s metasearch engine performs additional searches until it is able to have confidence that the returned result provides relevant results. Ex. 1011, ¶83.

360. This “computation of the confidence of the relevance prediction,” can be done using “similarity measures..., or machine learning methods such as neural networks or support vector machines.” Ex. 1011, ¶¶85-86.

361. By adopting Glover’s “selective retrieval,” Hartono’s ability for its Internet-focused task agents to search the Internet and return results would benefit from “improve[d] execution time, resource usage, and throughput by requiring fewer retrieval requests compared to content-based metasearch engines.” Ex. 1011, ¶5.

B. A POSITA Would Have Had A Reasonable Expectation of Success

362. A person of ordinary skill in the art would have easily been able to implement a metasearch engine for use by Hartono and to allow for Glover's dispatcher to act as Hartono's application task agents to invoke Internet searching. Further, there is nothing in Baker that would prevent the combination.

363. Glover's metasearch engine simply extends Hartono's Internet search functionality. As discussed Hartono already discusses providing Internet searching searches. A person of ordinary skill in the art would have implemented the details of the Internet searching functionality would also be motivated to improve that functionality by providing a robust system that could provide more relevant results by looking at multiple sources.

364. A person of ordinary skill in the art would have the skill to implement and the reasonable expectation of success in achieving such a combination because it would have involved a combination of known technologies (speech recognition and Internet searching based on user queries) according to known methods (speech recognition engines that convert speech into text strings and using text strings representing user's queries to perform Internet searching) to yield the predictable result of a method for inputting natural language speech and outputting a text string that can be used to invoke various applications to retrieve information from the Internet.

C. Dependent Claim 8

365. In my opinion, Hartono in view of Glover renders obvious dependent claim 8.

1. **Claim 8:** “*wherein one or more of the selected at least one of the plurality of domain agents is further configured to create duplicate queries and send the duplicate queries in an asynchronous manner to different information sources.*”

366. In my opinion, Hartono in view of Glover renders obvious this limitation.

367. In the combination, Hartono’s Internet-focused task agents (*domain agents*) would have employed Glover’s metasearch engine to create and send queries to different information sources based on the user’s input.

368. Specifically, Glover’s dispatcher would act as a task agent (*domain agent*) to send queries to multiple search engines (*different information sources*)

369. Glover discloses that as a general matter, “a metasearch engine takes user queries and submits them to multiple underlying search engines.” Ex. 1011, ¶35. It further discloses that its dispatcher is able to “choose search engines to query and...modify queries appropriately for each source.” *Id.*, ¶39.

370. A person of ordinary skill in the art would have understood that taking a single user query and submitting to multiple search engines discloses *creating duplicate queries and sending the duplicate queries to different information sources*,

as a query must be duplicated in order to be sent to multiple sources. Fig. 3 (showing “request submitter” sending duplicate queries to the World Wide Web).

371. A person of ordinary skill in the art would have also understood that Glover sends the queries to the World Wide Web (*external information sources*) in an *asynchronous manner*, as Glover states that its “steps may be performed in parallel,” “[f]or example, step 53 may be estimating the relevance of one or more results while step 52 is still sending queries 51 to and retrieving results 52 from search engines.” Ex. 1011, ¶82.

XIV. SECONDARY CONSIDERATIONS

372. I am not aware of any secondary considerations of non-obviousness for the claims of the ’825 patent. If Patent Owner raises any secondary considerations in its briefing or expert’s declaration, I reserve the right to respond to those considerations.

XV. CONCLUSION

373. For all the reasons stated above, it is my opinion that the challenged claims of the ’825 patent are unpatentable.

This declaration and my opinions are made to the best of my knowledge and understanding, and based on the material available to me at the time of signing this declaration. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true. Further, I am

Declaration of Daniel Lopresti Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 9,734,825

aware that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001. I declare under penalty of perjury that the foregoing is true and correct.

Executed this 15th day of August 2025 in Madrid, Spain.

A handwritten signature in black ink, appearing to read 'D. Lopresti', with a stylized flourish at the end.

Daniel P. Lopresti

Appendix A

Exhibit No.	Description
1001	U.S. Patent No. 9,734,825 to Kennewick, et al. (“the ’825 Patent”)
1002	File History of U.S. Patent No. 9,734,825
1003	Declaration of Daniel Lopresti (“Lopresti Decl.”)
1004	Curriculum Vitae of Daniel Lopresti
1005	U.S. Patent No. 7,085,723 to Ross et al. (“Ross”)
1006	WO 00/11571 to Hartono et al. (“Hartono”)
1007	U.S. Patent No. 5,937,383 to Ittycheriah et al. (“Ittycheriah”)
1008	U.S. Patent No. 5,680,511 to Baker et al. (“Baker”)
1009	U.S. Patent No. 7,249,018 to Ross et al. (“Ross018”)
1010	U.S. Patent No. 6,636,848 to Aridor et al. (“Aridor”)
1011	U.S. Patent Application Publication No. 2002/1065860 to Glover et al. (“Glover”)
1012	U.S. Patent No. 7,127,402 to Ross et al. (“Ross402”)
1013	Reserved
1014	Reserved
1015	Appendix E – Infringement Contentions for U.S. Patent No. 9,734,825, <i>Dialect LLC v. Meta Platforms, Inc.</i> , No. 7:25-CV-00060-DC-DTG (E.D. Tex. July 16, 2025)
1016	U.S. Provisional Patent Application No. 60/261372 to Ross et al.

Exhibit No.	Description
1017	IBM® ViaVoice® for Windows® Personal Edition, User's Guide, Release 10 (August 2002)
1018	Excerpts from IBM ViaVoice™ 98 Home Edition User Guide (July 1998)
1019	Excerpts from Frederick Jelinek, <i>Statistical Methods for Speech Recognition</i> , The MIT Press (1997)
1020	Michael McTear, <i>Spoken Dialogue Technology: Enabling the Conversational User Interface</i> , 34 ACM Computing Surveys No. 1, pp. 90-169 (Mar. 2002)
1021	Liam Comerford, et al., <i>The IBM Personal Speech Assistant</i> , IBM Watson Research Center, IEEE (2001)
1022	Eric J. Glover, et al., <i>Architecture of a Metasearch Engine that Supports User Information Needs</i> , In International Conference on Information and Knowledge Management, Proceedings, pp. 210-216, ACM (1999)
1023	Dragon Naturally Speaking® 5, User's Guide (August 2000)
1024	Definitions of “asynchronous” and “token” from Microsoft Computer Dictionary, Fourth Edition, Microsoft Corporation (1999)
1025	Definition of “dictionary” from Webster's Ninth New Collegiate Dictionary, Merriam-Webster Inc. (1990)
1026	Definition of “dictionary” from The American Heritage Dictionary of The English Language, Houghton Mifflin Co. (1992)