US011615145B2

(12) **United States Patent**
Galitsky

(10) **Patent No.: US 11,615,145 B2**
(45) **Date of Patent: Mar. 28, 2023**

(54) **CONVERTING A DOCUMENT INTO A CHATBOT-ACCESSIBLE FORM VIA THE USE OF COMMUNICATIVE DISCOURSE TREES**

(71) Applicant: **Oracle International Corporation**, Redwood Shores, CA (US)

(72) Inventor: **Boris Galitsky**, San Jose, CA (US)

(73) Assignee: **Oracle International Corporation**, Redwood Shores, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 114 days.

(21) Appl. No.: **16/939,531**

(22) Filed: **Jul. 27, 2020**

(65) **Prior Publication Data**

US 2020/0356605 A1 Nov. 12, 2020

Related U.S. Application Data

(63) Continuation-in-part of application No. 16/789,849, filed on Feb. 13, 2020, which is a continuation-in-part
(Continued)

(51) **Int. Cl.**
G06F 16/9032 (2019.01)
G06F 40/211 (2020.01)
(Continued)

(52) **U.S. Cl.**
CPC ***G06F 16/90332*** (2019.01); ***G06F 16/9027*** (2019.01); ***G06F 16/9538*** (2019.01); ***G06F 40/211*** (2020.01)

(58) **Field of Classification Search**
CPC G06F 16/90332; G06F 16/9027; G06F 16/9538; G06F 40/211; G06F 40/289; G06F 40/35; H04L 51/02; H04L 51/063
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,495,605	A	2/1996	Cadot
6,112,168	A	8/2000	Corston et al.

(Continued)

FOREIGN PATENT DOCUMENTS

WO	2015089822	6/2015

OTHER PUBLICATIONS

Seo et al., "Syntactic graphs: a representation for the union of all ambiguous parse trees" Computational Linguistics vol. 15 issue Mar. 1, 1989 pp. 19-32 (Year: 1989).*

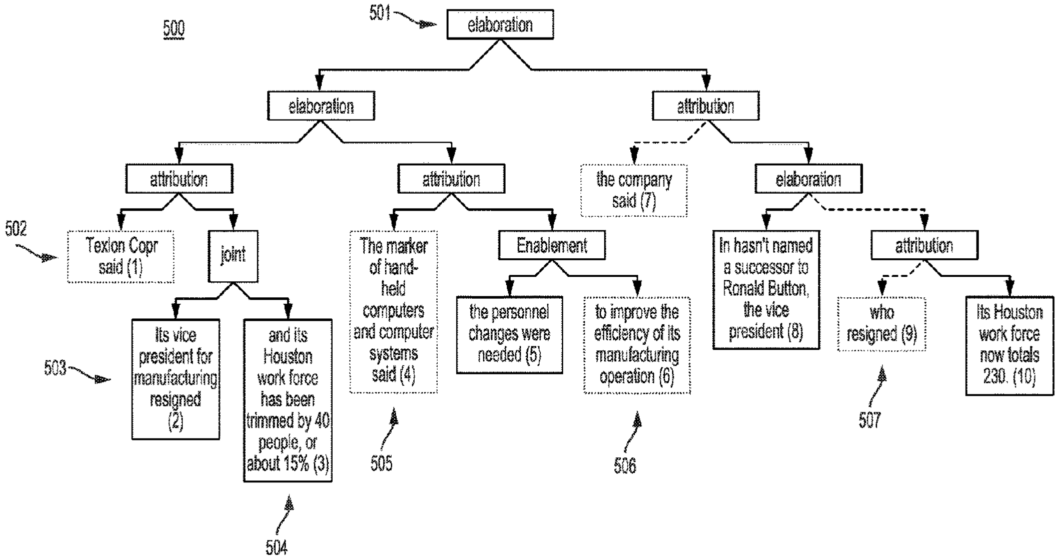
(Continued)

Primary Examiner — Mark D Featherstone
Assistant Examiner — Kweku William Halm
(74) *Attorney, Agent, or Firm* — Kilpatrick Townsend & Stockton LLP

(57) **ABSTRACT**

Techniques are disclosed for textual analysis. In an example, a text analysis application constructs a discourse tree that represents rhetorical relationships between fragments of text. The application matches each fragment that has a verb to a verb signature, thereby creating a communicative discourse tree. The application identifies, in the communicative discourse tree, a terminal node including a nucleus elementary discourse unit, which is dominant to a corresponding satellite elementary discourse unit. The application extracts, from the communicative discourse tree, a particular verb signature corresponding to the nucleus elementary discourse unit. The application determines a subject from the verb signature and forms a question fragment from the subject.

19 Claims, 22 Drawing Sheets


500
501
elaboration
elaboration
attribution
attribution
attribution
the company said (7)
elaboration
502
Texlon Copr said (1)
joint
The marker of hand-held computers and computer systems said (4)
Enablement
In hasn't named a successor to Ronald Button, the vice president (8)
attribution
503
Its vice president for manufacturing resigned (2)
and its Houston work force has been trimmed by 40 people, or about 15% (3)
the personnel changes were needed (5)
to improve the efficiency of its manufacturing operation (6)
who resigned (9)
Its Houston work force now totals 230. (10)
505
506
507
504

Related U.S. Application Data

of application No. 15/975,683, filed on May 9, 2018, now Pat. No. 10,796,102.

(60) Provisional application No. 62/894,162, filed on Aug. 30, 2019, provisional application No. 62/832,986, filed on Apr. 12, 2019, provisional application No. 62/504,377, filed on May 10, 2017.

(51) **Int. Cl.**

G06F 16/9538	(2019.01)
G06F 16/901	(2019.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,181,909	B1	1/2001	Burstein et al.	
6,731,307	B1	5/2004	Strubbe et al.	
7,152,031	B1	12/2006	Jensen et al.	
7,359,860	B1	4/2008	Miller et al.	
7,519,529	B1 *	4/2009	Horvitz	G06F 40/35 369/30.04
7,533,013	B2 *	5/2009	Marcu	G06F 40/35 704/7
7,551,552	B2	6/2009	Dunagan et al.	
7,840,556	B1	11/2010	Dayal et al.	
8,260,817	B2 *	9/2012	Boschee	G06F 16/334 707/794
9,037,464	B1	5/2015	Mikolov et al.	
9,292,490	B2	3/2016	Kimelfeld et al.	
9,559,993	B2	1/2017	Palakovich et al.	
9,582,501	B1	2/2017	Salmon et al.	
9,817,721	B1	11/2017	Youngworth	
10,019,716	B1	7/2018	Ainslie et al.	
10,175,865	B2	1/2019	Beaver et al.	
10,289,974	B1	5/2019	Ouimette	
10,545,648	B2	1/2020	Beaver et al.	
10,679,011	B2	6/2020	Galitsky	
10,796,102	B2	10/2020	Galitsky	
10,817,670	B2	10/2020	Galitsky	
10,839,154	B2	11/2020	Galitsky	
10,853,581	B2	12/2020	Galitsky	
11,023,684	B1	6/2021	Flor et al.	
11,100,144	B2	8/2021	Galitsky	
2001/0007987	A1	7/2001	Igata	
2001/0053968	A1	12/2001	Galitsky et al.	
2002/0040292	A1	4/2002	Marcu	
2002/0046018	A1	4/2002	Marcu et al.	
2003/0138758	A1	7/2003	Burstein et al.	
2004/0044519	A1 *	3/2004	Polanyi	G06F 40/35 707/E17.058
2004/0148170	A1	7/2004	Acero et al.	
2005/0086592	A1	4/2005	Polanyi et al.	
2007/0073533	A1	3/2007	Thione et al.	
2007/0136284	A1	6/2007	Cobb et al.	
2007/0192306	A1	8/2007	Papakonstantinou et al.	
2007/0294229	A1	12/2007	Au	
2008/0172409	A1	7/2008	Botros et al.	
2008/0228467	A1 *	9/2008	Womack	G06F 40/30 704/9
2009/0089252	A1	4/2009	Galitsky et al.	
2009/0100053	A1	4/2009	Boschee et al.	
2009/0248399	A1	10/2009	Au	
2009/0282019	A1	11/2009	Galitsky et al.	
2010/0169309	A1 *	7/2010	Barrett	G06F 16/951 707/E17.108
2010/0169359	A1	7/2010	Barrett et al.	
2010/0299136	A1	11/2010	Tong et al.	
2011/0119049	A1	5/2011	Ylonen	
2011/0153673	A1	6/2011	Boschee et al.	
2011/0231353	A1 *	9/2011	Wang	G06Q 10/10 706/45
2012/0041950	A1	2/2012	Koll et al.	
2012/0078902	A1	3/2012	Duboue et al.	
2012/0246578	A1	9/2012	Baldwin et al.	
2012/0290509	A1 *	11/2012	Heck	G10L 15/06 706/11
2013/0046757	A1	2/2013	Salvetti et al.	
2013/0151347	A1	6/2013	Baldwin et al.	
2013/0204611	A1	8/2013	Tsuchida et al.	
2013/0268532	A1	10/2013	Doshi	
2014/0040288	A1 *	2/2014	Galitsky	G06F 40/211 707/755
2014/0122083	A1 *	5/2014	Xiaojiang	H04L 51/02 704/270.1
2014/0136188	A1	5/2014	Wroczynski et al.	
2015/0039295	A1	2/2015	Soschen	
2015/0046492	A1	2/2015	Balachandran	
2015/0051900	A1	2/2015	Kimelfeld et al.	
2015/0134325	A1	5/2015	Skiba et al.	
2015/0149461	A1	5/2015	Aguilar Lemarroy et al.	
2015/0161512	A1 *	6/2015	Byron	G06F 16/24578 706/12
2016/0026608	A1	1/2016	Curin et al.	
2016/0034457	A1	2/2016	Bradley et al.	
2016/0055240	A1	2/2016	Tur et al.	
2016/0071517	A1	3/2016	Beaver et al.	
2016/0085743	A1	3/2016	Haley	
2016/0086601	A1	3/2016	Chotimongkol et al.	
2016/0099892	A1	4/2016	Palakovich et al.	
2016/0232152	A1	8/2016	Mahamood	
2016/0245779	A1	8/2016	Khalaj Amineh et al.	
2016/0246779	A1	8/2016	Ho et al.	
2016/0247068	A1 *	8/2016	Lin	G06F 40/40
2016/0283491	A1	9/2016	Lu et al.	
2016/0328667	A1	11/2016	Macciola et al.	
2017/0032053	A1	2/2017	LeTourneau	
2017/0104829	A1	4/2017	Degroat	
2017/0116982	A1	4/2017	Gelfenbeyn et al.	
2017/0177675	A1	6/2017	Beller et al.	
2017/0228368	A1	8/2017	Carter et al.	
2017/0277993	A1	9/2017	Beaver et al.	
2017/0286390	A1	10/2017	Yashpe et al.	
2017/0293651	A1 *	10/2017	Boguraev	G06N 20/00
2018/0121062	A1	5/2018	Beaver et al.	
2018/0181648	A1	6/2018	Chen	
2018/0189385	A1	7/2018	Sun et al.	
2018/0260472	A1 *	9/2018	Kelsey	G09B 7/02
2018/0314689	A1	11/2018	Wang et al.	
2018/0365228	A1	12/2018	Galitsky	
2018/0373701	A1	12/2018	McAteer et al.	
2019/0005027	A1	1/2019	He et al.	
2019/0033957	A1	1/2019	Shionozaki	
2019/0057157	A1	2/2019	Mandal et al.	
2019/0103111	A1	4/2019	Tiwari et al.	
2019/0138190	A1	5/2019	Beaver et al.	
2019/0163756	A1	5/2019	Bull et al.	
2019/0354544	A1	11/2019	Hertz et al.	
2019/0377605	A1	12/2019	Joseph	
2020/0099790	A1	3/2020	Ma et al.	
2020/0117858	A1	4/2020	Freeman et al.	
2020/0301589	A1	9/2020	Buzzard et al.	
2021/0020165	A1	1/2021	Scodary et al.	
2021/0027799	A1	1/2021	Scodary et al.	
2021/0029248	A1	1/2021	Scodary et al.	

OTHER PUBLICATIONS

Carlson et al., “Discourse lagging Reference manual” pp. 1-87 (Year: 2001).*

Joty et al., Combining Intra- and Multi-sentential Rhetorical Parsing for Document-level Discourse Analysis pp. 486-496 (Year: 2013).*

U.S. Appl. No. 16/789,849, Non-Final Office Action dated Feb. 17, 2022, 23 pages.

Galitsky et al., Style and Genre Classification by Means of Deep Textual Parsing, Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference Dialogue 2016, Jun. 2016, pp. 1-45.

Yao et al., Semantics-Based Question Generation and Implementation, Dialogue & Discourse, vol. 3, No. 2, Mar. 17, 2012, pp. 11-42.

2009 Annual Study: Global Cost of a Data Breach, Ponemon Institute, LLC, PGP, Apr. 2010, 36 pages.

(56) References Cited

OTHER PUBLICATIONS

Data Loss Prevention, Trend Micro, Available online at: http://www.trendmicro.co.in/cloud-content/us/pdfs/business/datasheets/ds_data-loss-prevention.pdf, 2010, 2 pages.

Data Loss Prevention & Protection, Symantec, Available online at: http://www.symantec.com/business/theme.jsp?themeid=vontu, Accessed from Internet on Aug. 30, 2018, 6 pages.

Exploring Dialog Management for Bots, Chatbots Magazine, Available online at: https://chatbotsmagazine.com/exploring-dialog management-for-bots-cbb8665a2fd3, Jul. 11, 2016, 7 pages.

Global Security Report 2010, Trustwave, Available online at: https://www.trustwave.com/Resources/Library/Documents/2010-Trustwave-Global-Security-Report/, 2010, 49 pages.

Ignore, Deny, Downplay: Accounts of Syrians from Douma Have No Place in Western Narrative, Russia Today, Available online at: https://www.rt.com/news/425438-douma-witnesses-gas-attack-syria/, Apr. 28, 2018, pp. 1-8.

Malaysia Airlines Flight 17, Wikipedia, Available Online at: https://en.wikipedia.org/wiki/Malaysia_Airlines_Flight 17, 2016, pp. 1-38.

Shadow Chairman of Investigative Committee, Crime Russia, Available online at: https://crimerussia.com/corruption/tenevoy-direktor-skr/, Aug. 25, 2016, 5 pages.

Welcome to Apache Lucene, Apache Lucene 7.5.0 and Apache Solr 7.5.0, Available online at: www.lucene.apache.org, Sep. 24, 2018, 38 pages.

U.S. Appl. No. 15/975,683, Non-Final Office Action dated Mar. 19, 2020, 16 pages.

U.S. Appl. No. 15/975,683, Non-Final Office Action dated Oct. 31, 2019, 27 pages.

U.S. Appl. No. 15/975,683, Notice of Allowance dated Jun. 12, 2020, 17 pages.

U.S. Appl. No. 15/975,685, Non-Final Office Action dated Apr. 1, 2020, 23 pages.

U.S. Appl. No. 15/975,685, Non-Final Office Action dated Nov. 15, 2019, 23 pages.

U.S. Appl. No. 15/975,685, Notice of Allowance dated Jul. 24, 2020, 17 pages.

U.S. Appl. No. 16/010,091, Non-Final Office Action dated Nov. 18, 2019, 26 pages.

U.S. Appl. No. 16/010,091, Notice of Allowance dated Mar. 19, 2020, 13 pages.

U.S. Appl. No. 16/010,141, Non-Final Office Action dated Feb. 24, 2020, 12 pages.

U.S. Appl. No. 16/010,156, Notice of Allowance dated Feb. 6, 2020, 13 pages.

U.S. Appl. No. 16/010,156, Notice of Allowance dated Nov. 7, 2019, 13 pages.

U.S. Appl. No. 16/145,702, Final Office Action dated May 6, 2020, 19 pages.

U.S. Appl. No. 16/145,702, Final Office Action dated Sep. 10, 2019, 25 pages.

U.S. Appl. No. 16/145,702, First Action Interview Office Action Summary dated Apr. 29, 2019, 8 pages.

U.S. Appl. No. 16/145,702, First Action Interview Pilot Program Pre-Interview Communication dated Feb. 7, 2019, 6 pages.

U.S. Appl. No. 16/145,702, Non-Final Office Action dated Feb. 5, 2020, 30 pages.

U.S. Appl. No. 16/145,702, Notice of Allowance dated Jul. 1, 2020, 15 pages.

U.S. Appl. No. 16/145,777, Non-Final Office Action dated Apr. 3, 2020, 18 pages.

U.S. Appl. No. 16/145,777, Notice of Allowance dated Jul. 15, 2020, 17 pages.

U.S. Appl. No. 16/260,939, Non-Final Office Action dated May 1, 2020, 10 pages.

U.S. Appl. No. 16/260,939, Notice of Allowance dated Jun. 12, 2020, 14 pages.

Airenti et al., Conversation and Behavior Games in the Pragmatics of Dialogue, Cognitive Science, vol. 17, No. 2, Apr.-Jun. 1993, pp. 197-256.

Allen et al., Analyzing Intention in Utterances, Artificial Intelligence, vol. 15, No. 3, Dec. 1980, pp. 143-178.

Banchs, Movie-DiC: a Movie Dialogue Corpus for Research and Development, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, vol. 2, Jul. 2012, pp. 203-207.

Bengio et al., A Neural Probabilistic Language Model, Journal of Machine Learning Research, vol. 3, Feb. 3, 2003, pp. 1137-1155.

Blaylock, Managing Communicative Intentions in Dialogue Using a Collaborative Problem-Solving Model, The University of Rochester, Computer Science Department, Technical Report 774, Apr. 2002, 56 pages.

Blaylock et al., Managing Communicative Intentions with Collaborative Problem Solving, Current and New Directions in Discourse and Dialogue, Chapter-4, 2003, pp. 63-84.

Boyer et al., MJRTY—A Fast Majority Vote Algorithm, Chapter 5, Automated Reasoning, 1991, pp. 105-117.

Carlson et al., Discourse Tagging Reference Manual, Available online at: https://www.isi.edu/~marcu/discourse/tagging-ref-manual.pdf, Sep. 11, 2001, 87 pages.

Chali et al., Complex Question Answering: Unsupervised Learning Approaches and Experiments, Journal of Artificial Intelligence Research, vol. 35, May 2009, pp. 1-47.

Chen, Understanding Mental States in Natural Language, Proceedings of the 8th International Conference on Computational Semantics, Jan. 2009, pp. 61-72.

Cohen, Enron Email Dataset, Available online at: https://www.cs.cmu.edu/~enron/, Jul. 10, 2016, 1 page.

Cohen et al., Intention is Choice with Commitment, Artificial Intelligence, vol. 42, Nos. 2-3, Mar. 1990, pp. 213-261.

Collins et al., New Ranking Algorithms for Parsing and Tagging: Kernels Over Discrete Structures, and The Voted Perceptron, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Jul. 2002, pp. 263-270.

Cristea et al., Veins Theory: A Model of Global Discourse Cohesion and Coherence, In C. Boitet & P. Whitelock (Eds.), 17th International Conference on Computational Linguistics, 1998, pp. 281-285.

Croft et al., Search Engines—Information Retrieval in Practice, Pearson Education, 2010, 542 pages.

De Boni, Using Logical Relevance for Question Answering, Journal of Applied Logic, vol. 5, No. 1, 2007, pp. 92-103.

De Mori et al., Spoken Language Understanding, Institute of Electrical and Electronics Engineers Signal Processing Magazine, vol. 25, No. 3, May 2008, pp. 50-58.

Dijkstra, Programming Considered as a Human Activity, Proc. IFIP Congress, 1965, 7 pages.

Du et al., Learning to Ask: Neural Question Generation for Reading Comprehension, Proceedings ofthe 55th Annual Meeting of the Association for Computational Linguistics, vol. 1, Apr. 29, 2017, 11 pages.

Ebrahim, NLP Tutorial Using Python NLTK (Simple Examples, Dzone, Available online at: https://dzone.com/articles/nlp-tutorial-using-python-nltk-simple-examples, Sep. 24, 2017, pp. 1-10.

Endres-Niggemeyer et al., Summarizing Text for Intelligent Communication, Dagstuhl Seminar Report 79, 1995, 36 pages.

Feng et al., A Linear-Time Bottom-Up Discourse Parser with Constraints and Post-Editing, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Jun. 23-25, 2014, pp. 511-521.

Feng et al., Classifying Arguments by Scheme, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, vol. 1, Jun. 19-24, 2011, pp. 987-996.

Feng et al., Distributional Footprints of Deceptive Product Reviews, Proceedings of the Sixth International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media, The Association for the Advancement of Artificial Intelligence Press, Jan. 2012, pp. 98-105.

Feng, RST-Style Discourse Parsing and Its Applications in Discourse Analysis, University of Toronto, Jun. 2015, 189 pages.

Feng et al., Syntactic Stylometry for Deception Detection, In Association for Computational Linguistics 12, Proceedings of the

(56) **References Cited**

OTHER PUBLICATIONS

50th Annual Meeting of the Association for Computational Linguistics, Jul. 8-14, 2012, pp. 171-175.

Ferraiolo et al., Role-Based Access Controls, Proceedings of the 15th NIST-NSA National Computer Security Conference, Oct. 13-16, 1992, pp. 554-563.

Finn, A Question Writing Algorithm, Journal of Reading Behavior, VII, vol. 4, 1975, pp. 341-367.

Fornaciari et al., Identifying Fake Amazon Reviews as Learning from Crowds, Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Apr. 26-30, 2014, pp. 279-287.

Galitsky et al., Chatbot with a Discourse Structure-Driven Dialogue Management, Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Apr. 3-7, 2017, pp. 87-90.

Galitsky et al., Concept-Based Learning of Human Behavior for Customer Relationship Management, Information Sciences, vol. 181, No. 10, May 15, 2011, pp. 2016-2035.

Galitsky et al., Discourse-level Dialogue Management, In Developing Enterprise Chatbots: Springer Nature, 2019, pp. 365-387.

Galitsky, Discovering Rhetorical Agreement between a Request and Response, Dialogue & Discourse, vol. 8, No. 2, Dec. 1, 2017, pp. 167-205.

Galitsky et al., Finding Maximal Common Sub-Parse Thickets for Multi-Sentence Search, Graph Structures for Knowledge Representation and Reasoning, 2014, 19 pages.

Galitsky et al., Improving Text Retrieval Efficiency with Pattern Structures on Parse Thickets, Formal Concept Analysis Meets Information Retrieval, ECIR, 2013, 16 pages.

Galitsky et al., Improving Trust in Automation of Social Promotion, Association for the Advancement of Artificial Intelligence, 2014, pp. 28-35.

Galitsky et al., Inferring the Semantic Properties of Sentences by Mining Syntactic Parse Trees, Data & Knowledge Engineering, vol. 81-82, Available online at: http://dx.doi.org/10.1016/j.datak.2012.07.003.2012, Nov.-Dec. 2012, 44 pages.

Galitsky et al., Learning Communicative Actions of Conflicting Human Agents, Journal of Experimental & Theoretical Artificial Intelligence, vol. 20, No. 4, Dec. 2008, pp. 277-317.

Galitsky, Learning Noisy Discourse Trees, Computational Linguistics and Intellectual Technologies, Proceedings of the International Conference "Dialogue 2017". Available online at: http://www.dialog-21.ru/media/3911/galitskyb.pdf, May 31-Jun. 3, 2017, 14 pages.

Galitsky, Learning Parse Structure of Paragraphs and its Applications in Search, Engineering Applications of Artificial Intelligence, vol. 32, Jun. 2014, pp. 160-184.

Galitsky, Machine Learning of Syntactic Parse Trees for Search and Classification of Text, Engineering Applications of Artificial Intelligence, vol. 26, Available Online at: http://dx.doi.org/10.1016/j.engappai.2012.09.017, Mar. 2013, pp. 1072-1091.

Galitsky, Matching Parse Thickets for Open Domain Question Answering, Data & Knowledge Engineering, vol. 107, Dec. 9, 2016, pp. 24-50.

Galitsky, Natural Language Question Answering System, Technique of Semantic Headers, Advanced Knowledge International, vol. 2, Apr. 2003, 333 pages.

Galitsky et al., "On a Chat Bot Finding Answers with Optimal Rhetoric Representation", RANLP—Recent Advances in Natural Language Processing Meet Deep Learning, Nov. 10, 2017, pp. 253-259.

Galitsky et al., Parse Thicket Representations for Answering Multi-Sentence Search, International Conference on Conceptual Structures, vol. 7735, 2013, pp. 153-172.

Galitsky et al., Rhetoric Map of an Answer to Compound Queries, Proceedings of the 53rd Annual Meeting of the 20 Association for Computational Linguistics and the 7th International Joint Conference of Natural Language Processing, vol. 2, Jul. 26-31, 2015, pp. 681-686.

Galitsky et al., Text Classification Based on Deep Textual Parsing, Available Online at: http://ceur-ws.org/Vol-1886/paper_8.pdf, 2011, 9 pages.

Galitsky et al., Text Classification into Abstract Classes Based on Discourse Structure, Proceedings of Recent Advances in Natural Language Processing, Sep. 2015, pp. 200-207.

Galitsky et al., Text Integrity Assessment: Sentiment Profile vs Rhetoric Structure, CICLing, Springer International Publishing, Apr. 2015, pp. 126-139.

Galitsky, Using Extended Tree Kernels to Recognize Metalanguage in Text, Studies in Computational Intelligence, Feb. 2017, 26 pages.

Grefenstette et al., Multi-Step Regression Learning for Compositional Distributional Semantics, Proceedings of the 10th International Conference on Computational Semantics, Mar. 2013, 11 pages.

Grefenstette, Towards a Formal Distributional Semantics: Simulating Logical Calculi with Tensors, University of Oxford, Apr. 2013, 10 pages.

Grosz et al., Attention, Intentions, and the Structure of Discourse, Computational Linguistics, vol. 12, No. 3, Jul.-Sep. 1986, pp. 175-204.

Grosz et al., Discourse Analysis, in Understanding Spoken Language, Elsevier North-Holland, 1978, pp. 234-268.

Hai et al., Deceptive Review Spam Detection via Exploiting Task Relatedness and Unlabeled Data, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Nov. 1-5, 2016, pp. 1817-1826.

Hara et al., Exploring Difficulties in Parsing Imperatives and Questions, Proceedings of the 5th International Joint Conference on Natural Language Processing, Nov. 8-13, 2011, pp. 749-757.

Hart et al., Text Classification for Data Loss Prevention, Proceedings of the 11th International Conference on Privacy Enhancing Technologies, Hewlett-Packard Development Company, L.P., Jul. 27-29, 2011, 21 pages.

Haussler, Convolution Kernels on Discrete Structures UCSC-CRL-99-10, University of California, Santa Barbara Technical Report, Jul. 8, 1999, 38 pages.

Heilman et al., Good Question! Statistical Ranking for Question Generation, Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Jun. 2010, pp. 609-617.

Hernault et al., A Sequential Model for Discourse Segmentation, International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2010: Computational Linguistics and Intelligent Text Processing, Mar. 21-27, 2010, pp. 315-326.

Houngbo et al., An Automated Method to Build A Corpus of Rhetorically-Classified Sentences in Biomedical Texts, Proceedings of the First Workshop on Argumentation Mining, Association for Computational Linguistics, Jun. 26, 2014, pp. 19-23.

Iruskieta et al., A Qualitative Comparison Method for Rhetorical Structures: Identifying Different Discourse Structures in Multilingual Corpora, Lang Resources & Evaluation, vol. 49, No. 2, May 8, 2014, 47 pages.

Jansen et al., Discourse Complements Lexical Semantics for Non-Factoid Answer Reranking, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1, Jun. 23-25, 2014, pp. 977-986.

Jasinskaja et al., Rhetorical Relations, Aug. 13, 2015, pp. 1-23.

John et al., Estimating Continuous Distributions in Bayesian Classifiers, Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995, pp. 338-345.

Johnson et al., The FrameNet Tagset for Frame-Semantic and Syntactic Coding of Predicate-Argument Structure, Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics, Apr. 2000, pp. 56-62.

Joty et al., A Novel Discriminative Framework for Sentence-Level Discourse Analysis, Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jul. 12-14, 2012, pp. 904-915.

Joty et al., CODRA: A Novel Discriminative Framework for Rhetorical Analysis, Computational Linguistics, vol. 41, No. 3, Mar. 18, 2015, pp. 385-435.

(56) **References Cited**

OTHER PUBLICATIONS

Joty et al., Combining Intra- and Multi-Sentential Rhetorical Parsing for Document-Level Discourse Analysis, 51st Annual Meeting of the Association for Computational Linguistics, vol. 1, Aug. 4-9, 2013, pp. 486-496.

Joty et al., Discriminative Reranking of Discourse Parses Using Tree Kernels, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Oct. 25-29, 2014, pp. 2049-2060.

Jurafsky et al., "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", Pearson, Prentice Hall, 2000, pp. 719-761.

Kate et al., Learning to Transform Natural to Formal Languages, Conference: Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, Jul. 2005, pp. 1062-1068.

Kipper et al., A Large-Scale Classification of English Verbs, Language Resources and Evaluation, vol. 42, No. 1, Mar. 2008, pp. 21-40.

Kipper et al., VerbNet Overview, Extensions, Mappings and Applications, Proceedings of Human Language Technologies, Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion, Volume: Tutorial Abstracts, Jun. 2009, pp. 13-14.

Kirschner et al., Linking the Thoughts: Analysis of Argumentation Structures in Scientific Publications, Proceedings of the 2nd Workshop on Argumentation Mining, Jun. 4, 2015, pp. 1-11.

Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence, vol. 2, Aug. 1995, pp. 1137-1143.

Kong et al., Improve Tree Kernel-Based Event Pronoun Resolution with Competitive Information, Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, vol. 3, Jul. 16-22, 2011, pp. 1814-1819.

Kontos et al., Question Answering and Rhetoric Analysis of Biomedical Texts in the Aroma System, National and Kapodistrian University of Athens, Unpublished Manuscript, 2006, 6 pages.

Kumar et al., A Framework for Automatic Question Generation from Text using Deep Reinforcement Learning, Proceedings of the 2019 IJCAI Workshop SCAI: The 4th International Workshop on Search-Oriented Conversational AI, Aug. 3, 2019, 7 pages.

Levinson, Presumptive Meanings: The Theory of Generalized Conversational Implicature, Cambridge, MA: The Massachusetts Institute of Technology Press, 2000, 10 pages.

Li et al., Recursive Deep Models for Discourse Parsing, Conference: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Jan. 2014, 10 pages.

Lin et al., Recognizing Implicit Discourse Relations in the Penn Discourse Treebank, Conference: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP, Aug. 6-7, 2009, pp. 343-351.

Litman et al., A Plan Recognition Model for Subdialogues in Conversations, Cognitive Science, vol. 11, No. 2, Apr. 1987, pp. 163-200.

Liu et al., How Not to Evaluate Your Dialogue System: an Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation, Proceedings of the Conference on Empirical Methods in Natural Language Processing, Jan. 3, 2017, 15 pages.

Makhalova et al., Pattern Structures for News Clustering, Proceedings of the 4th International Conference on What can FCA do for Artificial Intelligence, vol. 1430, Jul. 2015, pp. 35-42.

Mann et al., Discourse Structures for Text Generation, Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics, Jul. 2-6, 1984, pp. 367-375.

Mann et al., Rhetorical Structure Theory and Text Analysis, University of Southern California, Nov. 1989, 66 pages.

Mann et al., Rhetorical Structure Theory and Text Analysis, Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text, Apr. 8, 1992, pp. 39-78.

Mann et al., Rhetorical Structure Theory: Toward a Functional Theory of Text Organization, Text-Interdisciplinary Journal for the Study of Discourse, vol. 8, No. 3, Jan. 1988, pp. 243-281.

Marcu et al., An Unsupervised Approach to Recognizing Discourse Relations, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, 8 pages.

Mercier et al., Why Do Humans Reason? Arguments for an Argumentative Theory, Behavioral and Brain Sciences, vol. 34, No. 2, Apr. 2011, pp. 57-111.

Mikolov et al., Distributed Representations of Phrases and Their Compositionality, Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, Dec. 5-10, 2013, pp. 1-9.

Mikolov et al., Distributed Representations of Words and Phrases and Their Compositionality, Advances in Neural Information Processing Systems, vol. 26, Oct. 2013, 9 pages.

Mitchell et al., Composition in Distributional Models of Semantics, Cognitive Science, vol. 34, No. 8, Mar. 25, 2010, pp. 1388-1429.

Mitocariu et al., Comparing Discourse Tree Structures, Computational Linguistics and Intelligent Text Processing 14th International Conference, vol. 7816, Mar. 24-30, 2013, 11 pages.

Mochales et al., Argumentation Mining, Artificial Intelligence and Law, vol. 19, No. 1, Mar. 2011, pp. 1-22.

Mukherjee et al., Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews, Technical Report, Department of Computer Science, 2013, 11 pages.

Mukherjee et al., What Yelp Fake Review Filter Might Be Doing?, Proceedings of the Seventh International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media, Jan. 2013, pp. 409-418.

Nakamura et al., Another Diversity-Promoting Objective Function for Neural Dialogue Generation, Association for the Advancement of Artificial Intelligence, Nov. 21, 2018, 8 pages.

Novikova et al., Why We Need New Evaluation Metrics for NLG, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Sep. 7-11, 2017, pp. 2241-2252.

Ott et al., Finding Deceptive Opinion Spam by Any Stretch of the Imagination, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, vol. 1, Jun. 19-24, 2011, pp. 309-319.

Ott et al., Negative Deceptive Opinion Spam, Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics, Jun. 9-14, 2013, pp. 497-501.

Papineni et al., BLEU: A Method for Automatic Evaluation of Machine Translation, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Jul. 2002, pp. 311-318.

International Application No. PCT/US2018/031890, International Preliminary Report on Patentability dated Nov. 21, 2019, 9 pages.

International Application No. PCT/US2018/053392, International Preliminary Report on Patentability dated Apr. 9, 2020, 7 pages.

International Application No. PCT/US2018/053392, International Search Report and Written Opinion dated Dec. 17, 2018, 11 pages.

International Application No. PCT/US2019/015696, International Search Report and Written Opinion dated Apr. 23, 2019, 12 pages.

International Application No. PCT/US2019/031580, International Search Report and Written Opinion dated Jul. 5, 2019, 12 pages.

Peldszus et al., From Argument Diagrams to Argumentation Mining in Texts: A Survey, International Journal of Cognitive Informatics and Natural Intelligence, vol. 7, No. 1, Jan. 2013, pp. 1-31.

Pendyala et al., Towards a Truthful World Wide Web from a Humanitarian Perspective, Institute of Electrical and Electronics Engineers 2015 Global Humanitarian Technology Conference, Oct. 8-11, 2015, 7 pages.

Ponti, Machine Learning Techniques Applied to Dependency Parsing, Available online at: https://vision.unipv.it/AI/AIRG/MachineLearningTechniquesAppliedToDependencyParsingRevised.pdf, Oct. 2015, 45 pages.

(56) References Cited

OTHER PUBLICATIONS

Popescu et al., Logic-Based Rhetorical Structuring for Natural Language Generation in Human-Computer Dialogue, Proceedings of the 10th International Conference on Text, Speech and Dialogue, Sep. 3-7, 2007, pp. 309-317.

Popescu-Belis, Dialogue Acts: One or More Dimensions?, ISSCO Working Paper No. 62, University of Geneva, Nov. 2005, 46 pages.

Radev, A Common Theory of Information Fusion from Multiple Text Sources Step One: Cross-document Structure, Proceedings of the 1st SIGDial Workshop on Discourse and Dialogue, vol. 10, Oct. 7-8, 2000, pp. 74-83.

Radev et al., Centroid-Based Summarization of Multiple Documents: Sentence Extraction, Utility-Based Evaluation, 60 and User Studies, Proceedings of the NAACL-ANLP Workshop on Automatic Summarization, vol. 4, Dec. 2000, 10 pages.

Rayana et al., Collective Opinion Spam Detection: Bridging Review Networks and Metadata, Proceedings of the 21st Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining International Conference on Knowledge Discovery and Data Mining, Aug. 11-14, 2015, 10 pages.

Recasens et al., The Life and Death of Discourse Entities: Identifying Singleton Mentions, Proceedings of NAACL-HLT, Jun. 9-14, 2013, pp. 627-633.

Redeker, Coherence and Structure in Text and Discourse, In: William Black & Harry Bunt (eds.), Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics, Jan. 2000, pp. 1-28.

Reed et al., Language Resources for Studying Argument, Proceedings of the 6th Conference on Language Resources and Evaluation, LREC2008, ELRA, May 2010, pp. 2613-2618.

Reichman, Getting Computers to Talk Like You and Me, Discourse Context, Focus and Semantics (An ATN Model), Massachusetts Institute of Technology Press, Jul. 1985, pp. 35-49.

Salton et al., On the Specification of Term Values in Automatic Indexing, Journal of Documentation, vol. 29, No. 4, Apr. 1973, pp. 351-372.

Salton et al., Term Weighting Approaches in Automatic Text Retrieval, Information Processing & Management, vol. 24, No. 5, Nov. 1987, 22 pages.

Santhosh et al., Discourse Based Advancement on Question Answering System, International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), vol. 1, No. 2, Oct. 2012, 12 pages.

Scholman et al., A Step-Wise Approach to Discourse Annotation: Towards a Reliable Categorization of Coherence 64 Relations, Categories of Coherence Relations in Discourse Annotation, Dialogue & Discourse, vol. 7, No. 2, Feb. 2016, 28 pages.

Searle, Speech Acts: An Essay in the Philosophy of Language, Cambridge University Press, Jan. 1969, pp. 22-53.

Serban et al., A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues, Proceedings of the Thirty-First Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, Jun. 14, 2016, 15 pages.

Serban et al., Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models, Proceedings of the Thirtieth Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, Apr. 6, 2016, 8 pages.

Serban et al., Generating Factoid Questions with Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1, May 29, 2016, 13 pages.

Severyn et al., Fast Support Vector Machines for Convolution Tree Kernels, Data Mining Knowledge Discovery, vol. 25, No. 2, Sep. 2012, 33 pages.

Sjoera, The Linguistics Behind Chat Bots, iCapps, Available online at: http://www.icapps.com/the-linguistics-behind-chatbots/, Feb. 22, 2017, 9 pages.

Socher et al., Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks, Proceedings of the NIPS Deep Learning and Unsupervised Feature Learning Workshop, Jan. 2010, 9 pages.

Sperber et al., Relevance: Communication and Cognition, Blackwell, Oxford and Harvard University Press, Cambridge, MA, 1986, 331 pages.

Sun et al., Exploiting Product Related Review Features for Fake Review Detection, Mathematical Problems in Engineering, vol. 2016, Article ID 4935792, Jul. 4, 2016, 7 pages.

Sun et al., Exploring Syntactic Structural Features for Sub-Tree Alignment Using Bilingual Tree Kernels, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Jul. 2010, pp. 306-315.

Surdeanu et al., Two Practical Rhetorical Structure Theory Parsers, Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics-Human Language Technologies, Jun. 5, 2015, pp. 1-5.

Tai et al., Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks, Available Online at: https://arxiv.org/pdf/1503.00075.pdf, May 30, 2015, 11 pages.

Traum et al., Conversation Acts in Task-Oriented Spoken Dialogue, University of Rochester Computer Science, Computational Intelligence, vol. 8, No. 3, Aug. 1992, 31 pages.

Traum et al., Discourse Obligations in Dialogue Processing, Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, Jun. 27-30, 1994, pp. 1-8.

Traum, Rhetorical Relations, Action and Intentionality in Conversation, Proceedings ACL SIG Workshop on Intentionality and Structure in Discourse Relations, Jun. 1993, pp. 132-135.

Trstenjak et al., KNN with TF-IDF Based Framework for Text Categorization, Procedia Engineering, vol. 69, 24th Danube Adria Association for Automation and Manufacturing International Symposium on Intelligent Manufacturing and Automation, 2014, pp. 1356-1364.

Tsui, English Conversation. Describing English Language, Oxford University Press, 1994, 37 pages.

Uliyar, A Primer: Oracle Intelligent Bots, Powered by Artificial Intelligence, White Paper, Sep. 2017, 28 pages.

Van Der Wees et al., Five Shades of Noise: Analyzing Machine Translation Errors in User-Generated Text, Proceedings of the ACL 2015 Workshop on Noisy User-generated Text, Jul. 31, 2015, pp. 28-37.

Vapnik, The Nature of Statistical Learning Theory, Springer Science, 1995, 201 pages.

Walker et al., Quantitative and Qualitative Evaluation of Darpa Communicator Spoken Dialogue Systems, Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, Jul. 6-11, 2001, pp. 515-522.

Wang et al., A Joint Model for Question Answering and Question Generation, Learning to Generate Natural Language Workshop, ICML, Jun. 5, 2017, 7 pages.

Wang et al., Kernel Based Discourse Relation Recognition with Temporal Ordering Information, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Jul. 11-16, 2010, pp. 710-719.

Wang et al., Using Learning Analytics to Understand the Design of an Intelligent Language Tutor-Chatbot Lucy, International Journal of Advanced Computer Science and Applications, vol. 4, No. 11, Nov. 2013, pp. 124-131.

Wu et al., Enhancing Text Representation for Classification Tasks with Semantic Graph Structures, International Journal of Innovative Computing, Information and Control, vol. 7, No. 5 (B), May 2011, pp. 2689-2698.

Wuchner et al., Data Loss Prevention Based on Data-Driven Usage Control, Proceedings of the Institute of Electrical and Electronics Engineers 23rd International Symposium on Software Reliability Engineering, Nov. 27-30, 2012, pp. 151-160.

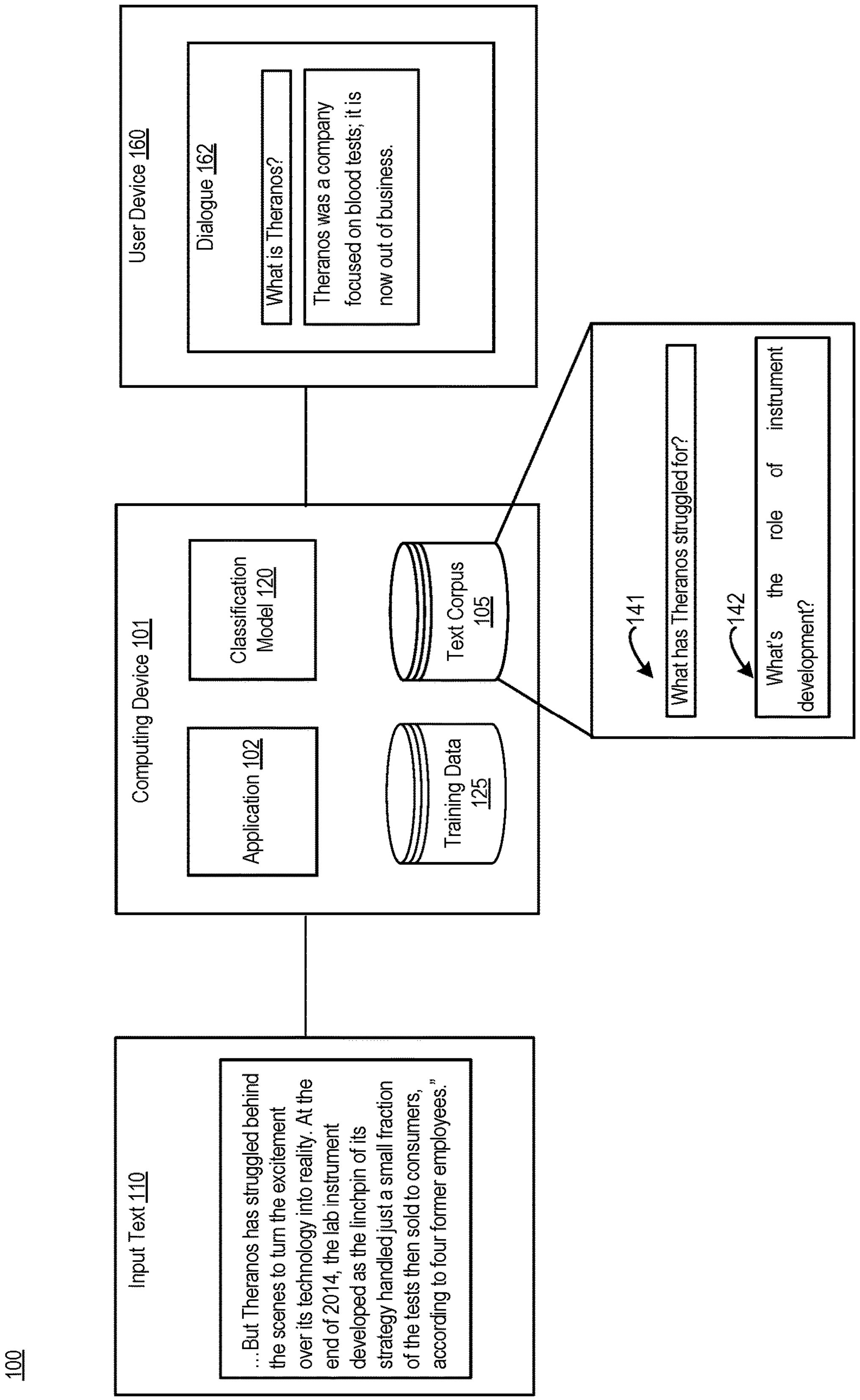
Yao et al., Online Deception Detection Refueled by Real World Data Collection, Proceedings of Recent Advances in Natural Language Processing, Jul. 28, 2017, 10 pages.

(56) **References Cited**

OTHER PUBLICATIONS

Yessenalina et al., Compositional Matrix-Space Models for Sentiment Analysis, Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Jul. 27-31, 2011, pp. 172-182.
Yuan et al., Machine Comprehension by Text-to-Text Neural Question Generation, Proceedings of the 2nd Workshop on Representation Learning for Natural Language Processing, Aug. 3, 2017, pp. 15-25.
Zanzotto et al., Estimating Linear Models for Compositional Distributional Semantics, Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Aug. 23-27, 2010, pp. 1263-1271.
Zeilenga, Lightweight Directory Access Protocol (LDAP) Read Entry Controls, Standards Track, Network Working Group, IETF, RFC 4527, Jun. 2006, 8 pages.
Zhao et al., Facilitating Discourse Analysis with Interactive Visualization, Institute of Electrical and Electronics Engineers Transactions on Visualization and Computer Graphics, vol. 18, No. 12, Dec. 2012, 10 pages.
International Application No. PCT/US2019/015696, International Preliminary Report on Patentability dated Aug. 13, 2020, 8 pages.
U.S. Appl. No. 16/789,849, Non-Final Office Action dated Jul. 1, 2022, 25 pages.
U.S. Appl. No. 16/145,777, Supplemental Notice of Allowability dated Sep. 2, 2020, 12 pages.
U.S. Appl. No. 16/736,517, Non-Final Office Action dated Dec. 8, 2021, 17 pages.
U.S. Appl. No. 16/736,517, Notice of Allowance dated Feb. 10, 2022, 11 pages.
U.S. Appl. No. 16/789,849, Notice of Allowance dated Oct. 26, 2022, 11 pages.
U.S. Appl. No. 16/822,563, Notice of Allowance dated Mar. 2, 2022, 16 pages.
U.S. Appl. No. 16/841,200, Notice of Allowance dated Apr. 14, 2022, 13 pages.
Appel et al., A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level, Knowledge-Based Systems, vol. 108, May 20, 2016, 32 pages.
Ariely, Predictably Irrational, 2008, 308 pages.
Biswa, Putt's Law, Peter Principle, Dilbert Principle of Incompetence & Parkinson's Law, Available Online at: http://asmilingassasin.blogspot.com/2015/06/putts-law-peter-principle-dilbert.html, Jun. 10, 2015, 5 pages.
Craig et al., Overhearing Dialogues and Monologues in Virtual Tutoring Sessions: Effects on Questioning and Vicarious Learning, International Journal of Artificial Intelligence in Education, Jan. 2000, pp. 242-253.
European Application No. 18727946.8, Office Action dated Dec. 15, 2021, 7 pages.
U.S. Appl. No. 16/998,915, Non-Final Office Action, dated Jan. 31, 2023, 27 pages.
Friederici et al., "Verb Argument Structure Processing: The Role of Verb-Specific and Argument-Specific Information", Journal Of Memory And Language, vol. 43, No. 3, Oct. 2000, pp. 476-507.
Pechsiri et al., "Mining Causality from Texts for Question Answering System", IEICE Transactions, On Information And Systems, vol. 90, No. 10, Oct. 2007, pp. 1523-1533.

* cited by examiner



…But Theranos has struggled behind the scenes to turn the excitement over its technology into reality. At the end of 2014, the lab instrument developed as the linchpin of its strategy handled just a small fraction of the tests then sold to consumers, according to four former employees."

What has Theranos struggled for?

What's the role of instrument development?

What is Theranos?

Theranos was a company focused on blood tests; it is now out of business.

FIG. 1

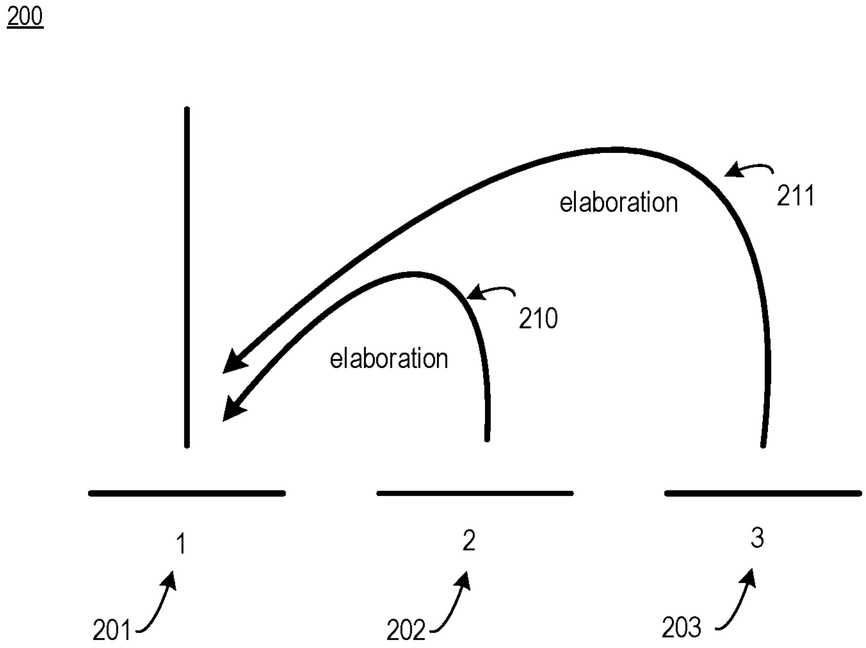


FIG. 2

FIG. 2

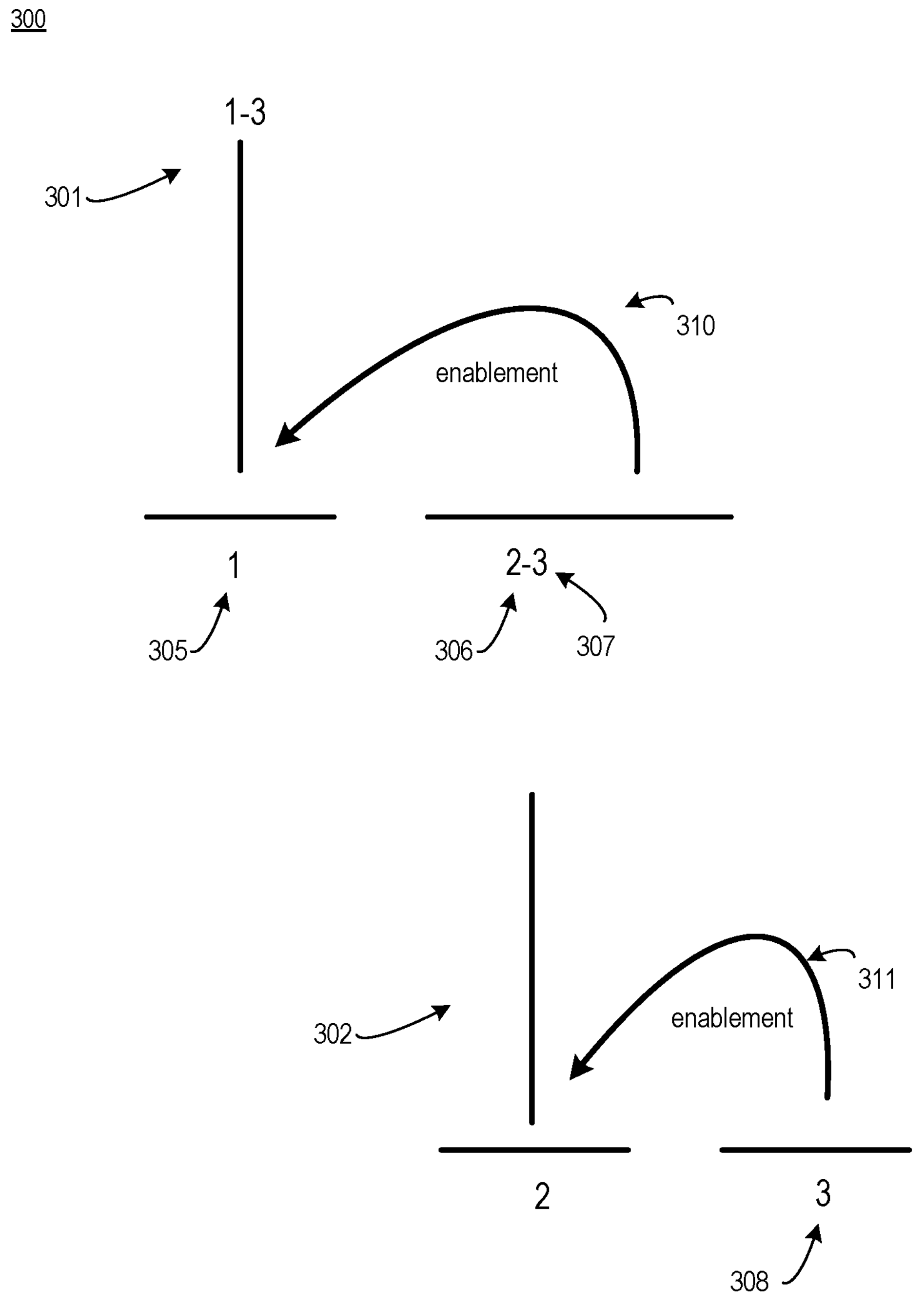

300
1-3
301
310
enablement
1
2-3
305
306
307
302
311
enablement
2
3
308

FIG. 3

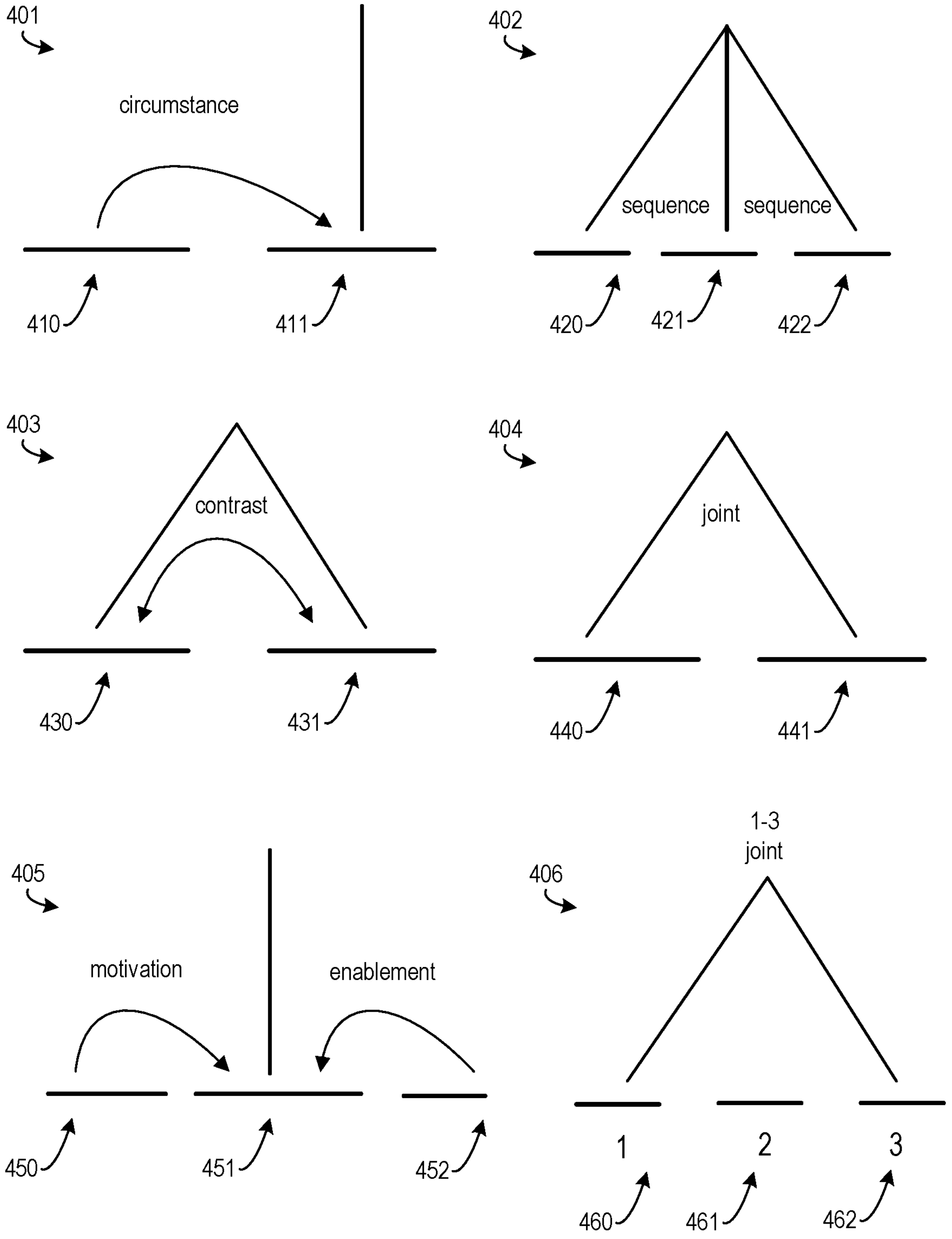

401
circumstance
410
411
402
sequence
sequence
420
421
422
403
contrast
430
431
404
joint
440
441
405
motivation
enablement
450
451
452
406
1-3
joint
1
2
3
460
461
462

FIG. 4

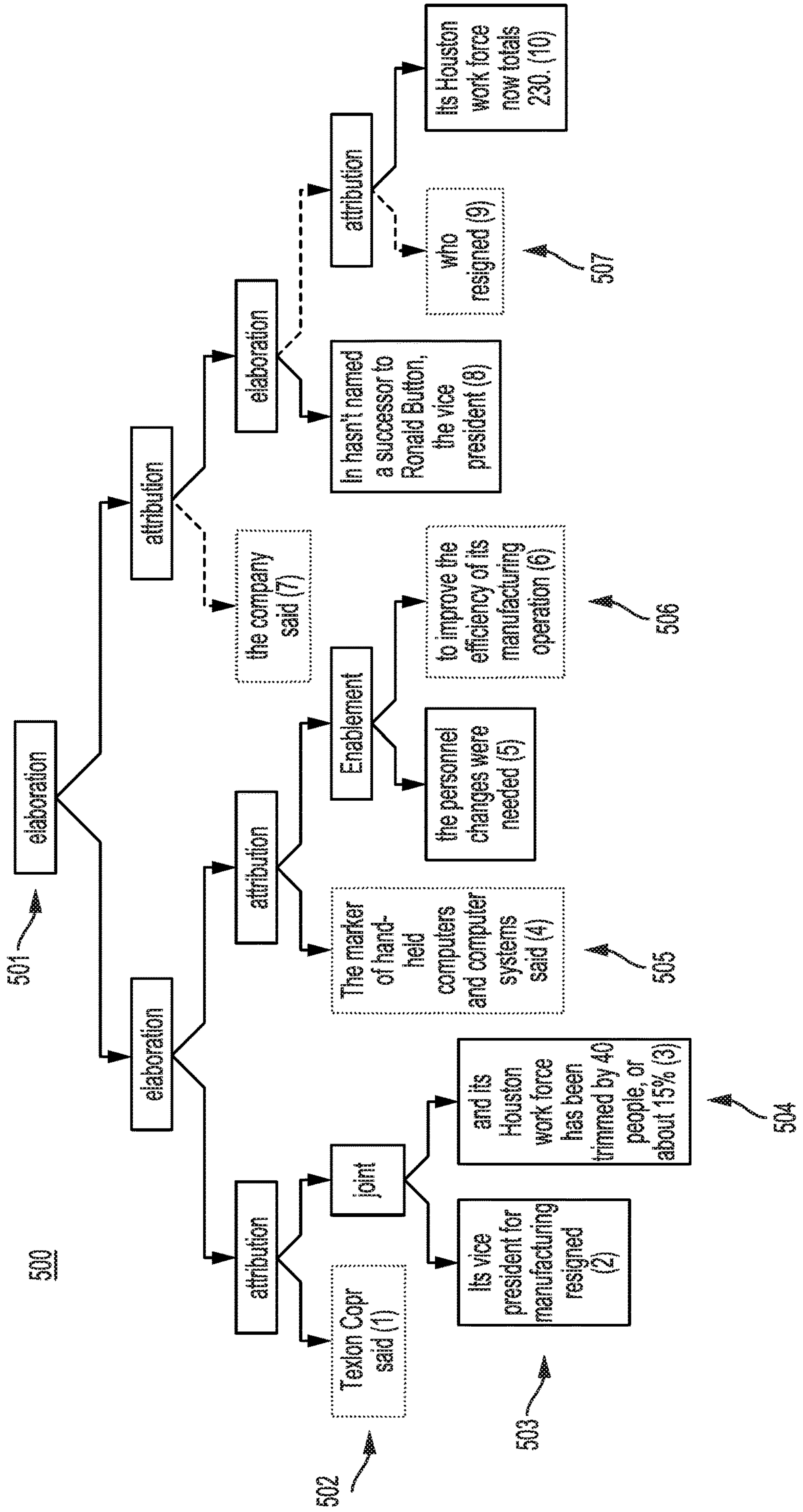
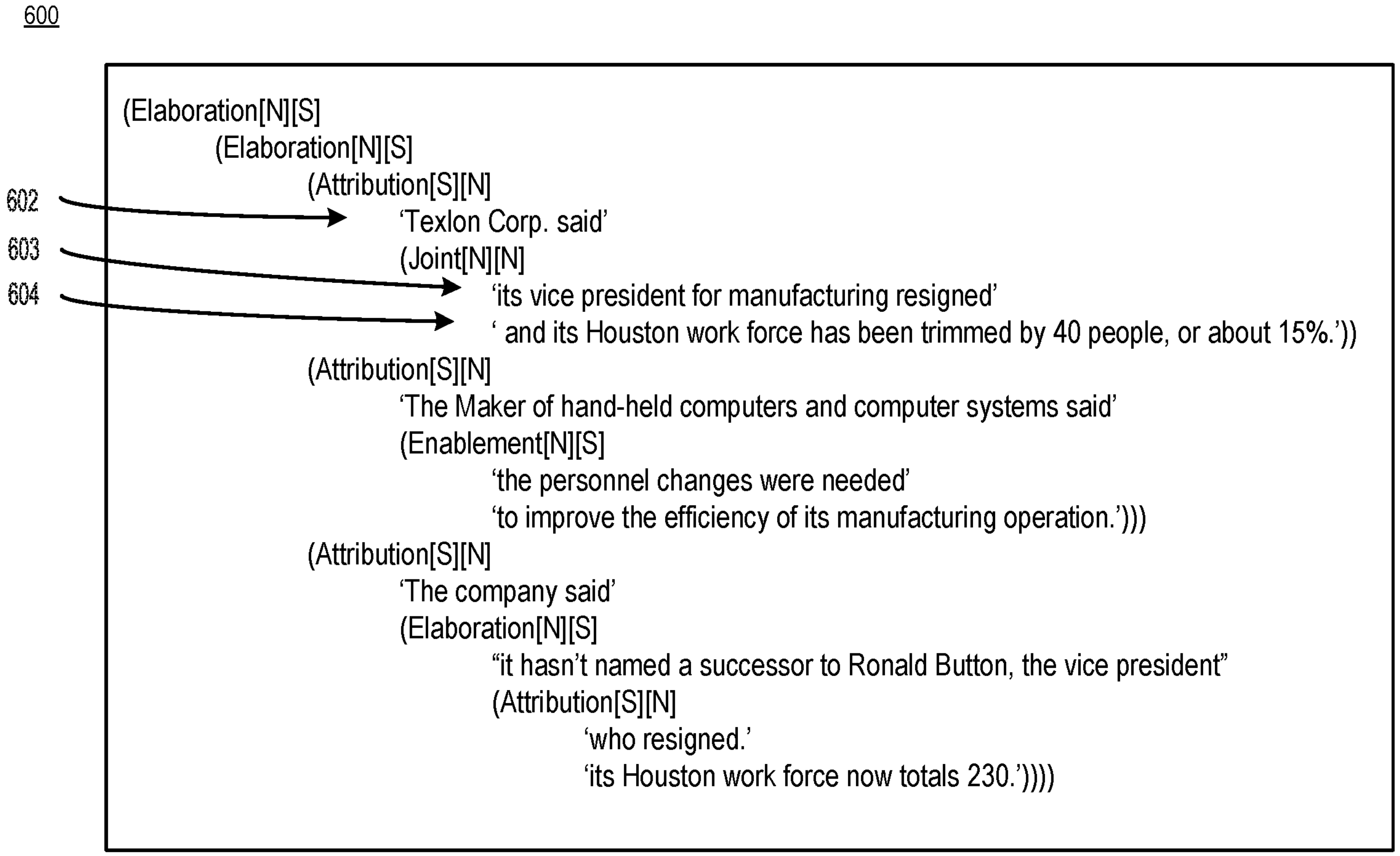

500
501
elaboration
elaboration
attribution
attribution
elaboration
Texlon Copr said (1)
502
joint
Its vice president for manufacturing resigned (2)
503
and its Houston work force has been trimmed by 40 people, or about 15% (3)
504
attribution
The marker of hand-held computers and computer systems said (4)
505
Enablement
the personnel changes were needed (5)
to improve the efficiency of its manufacturing operation (6)
506
the company said (7)
In hasn't named a successor to Ronald Button, the vice president (8)
attribution
who resigned (9)
507
Its Houston work force now totals 230. (10)

FIG. 5



```
(Elaboration[N][S]
        (Elaboration[N][S]
                (Attribution[S][N]
                        'Texlon Corp. said'
                        (Joint[N][N]
                                'its vice president for manufacturing resigned'
                                ' and its Houston work force has been trimmed by 40 people, or about 15%.'))
                (Attribution[S][N]
                        'The Maker of hand-held computers and computer systems said'
                        (Enablement[N][S]
                                'the personnel changes were needed'
                                'to improve the efficiency of its manufacturing operation.')))
                (Attribution[S][N]
                        'The company said'
                        (Elaboration[N][S]
                                "it hasn't named a successor to Ronald Button, the vice president"
                                (Attribution[S][N]
                                        'who resigned.'
                                        'its Houston work force now totals 230.'))))
```

FIG. 6

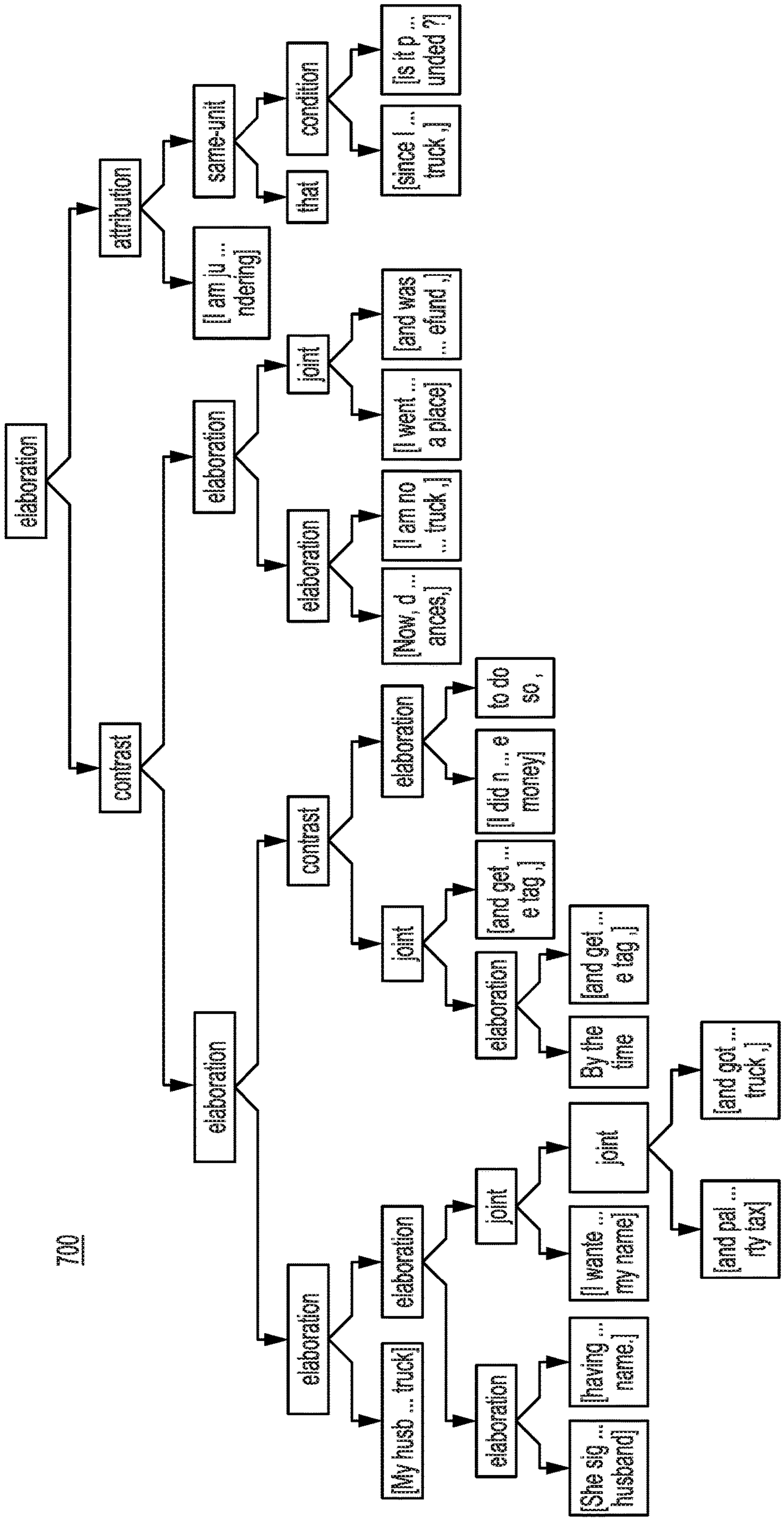

700
elaboration
contrast
attribution
elaboration
elaboration
[I am ju ... ndering]
same-unit
elaboration
contrast
elaboration
joint
that
condition
[My hus b ... truck]
elaboration
joint
elaboration
[Now, d ... ances,]
[I am no ... truck ,]
[I went ... a place]
[and was ... efund ,]
[since I ... truck ,]
[is it p ... unded ?]
elaboration
joint
elaboration
[and get ... e tag ,]
[I did n ... e money]
to do so ,
[She sig ... husband]
[having ... name,]
[I wante ... my name]
joint
By the time
[and get ... e tag ,]
[and pai ... rty tax]
[and got ... truck ,]

FIG. 7

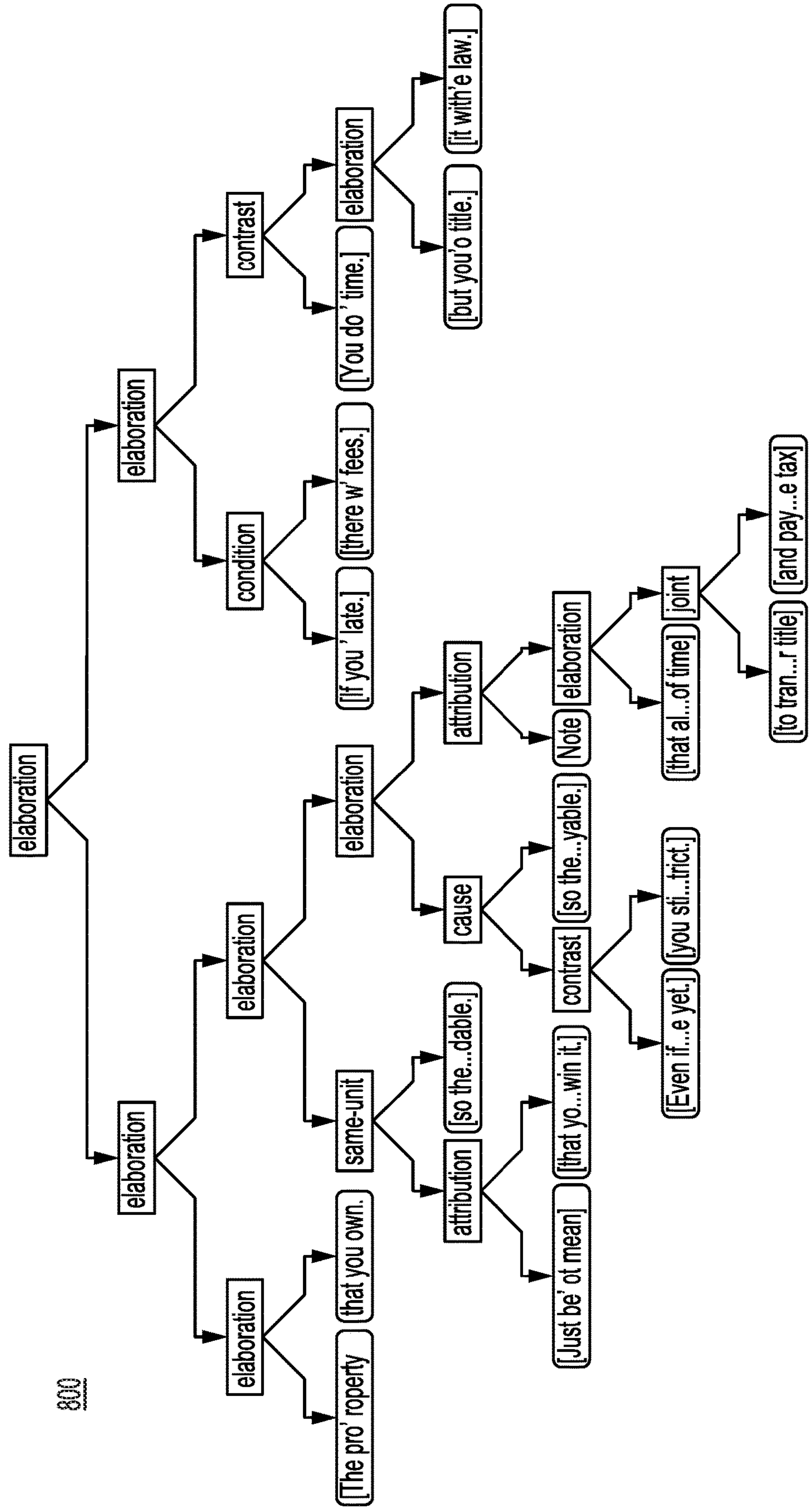

800
elaboration
elaboration
elaboration
[The pro' roperty
that you own.
elaboration
same-unit
attribution
[Just be' ot mean]
[that yo...win it,]
[so the...dable.]
elaboration
cause
contrast
[Even if...e yet.]
[you sti...trict.]
[so the...yable.]
attribution
Note
elaboration
[that al...of time]
joint
[to tran...r title]
[and pay...e tax]
elaboration
condition
[If you ' late.]
[there w' fees.]
contrast
[You do ' time.]
elaboration
[but you'o title.]
[it with'e law.]

FIG. 8

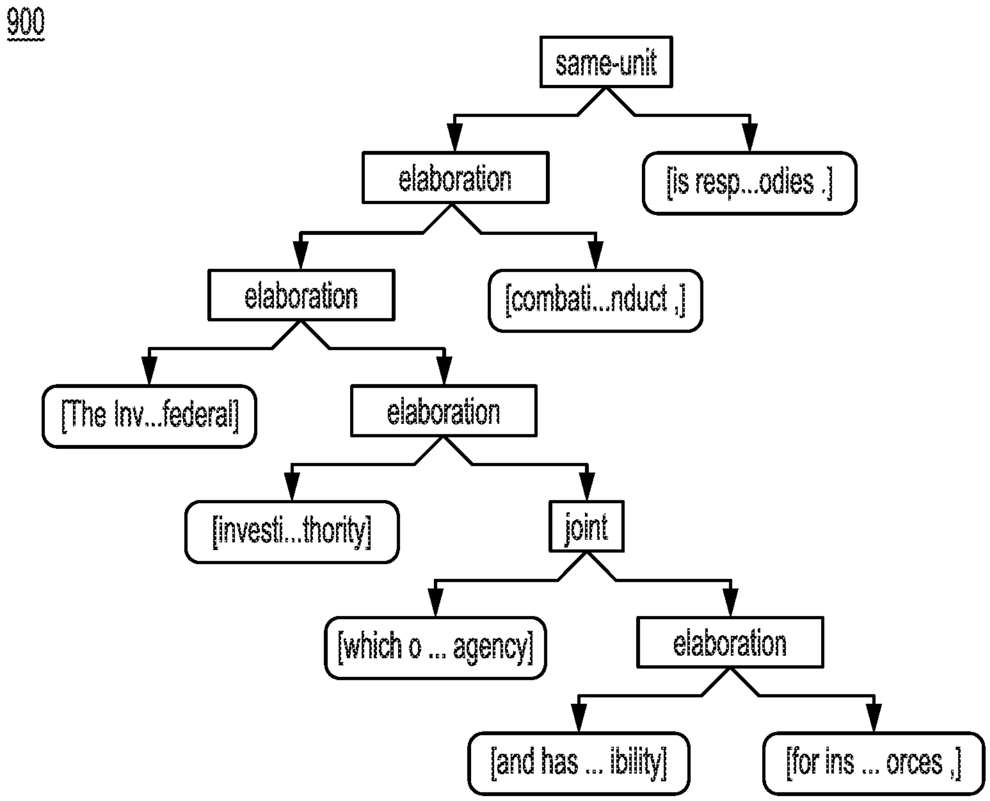

900
same-unit
elaboration
[is resp...odies .]
elaboration
[combati...nduct ,]
[The Inv...federal]
elaboration
[investi...thority]
joint
[which o ... agency]
elaboration
[and has ... ibility]
[for ins ... orces ,]

FIG. 9

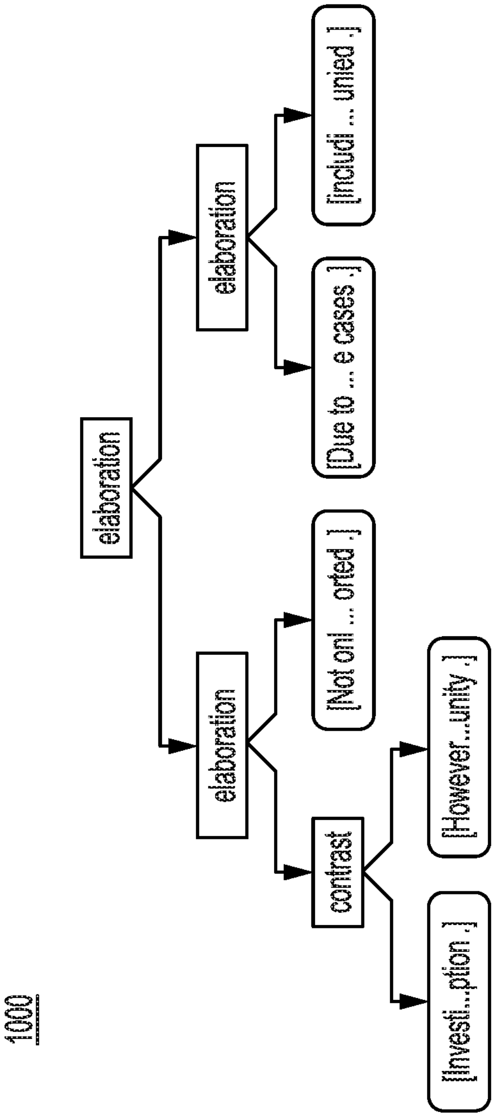

1000
elaboration
elaboration
elaboration
contrast
[Investi...ption .]
[However...unity .]
[Not onl ... orted .]
[Due to ... e cases .]
[includi ... unied .]

FIG. 10

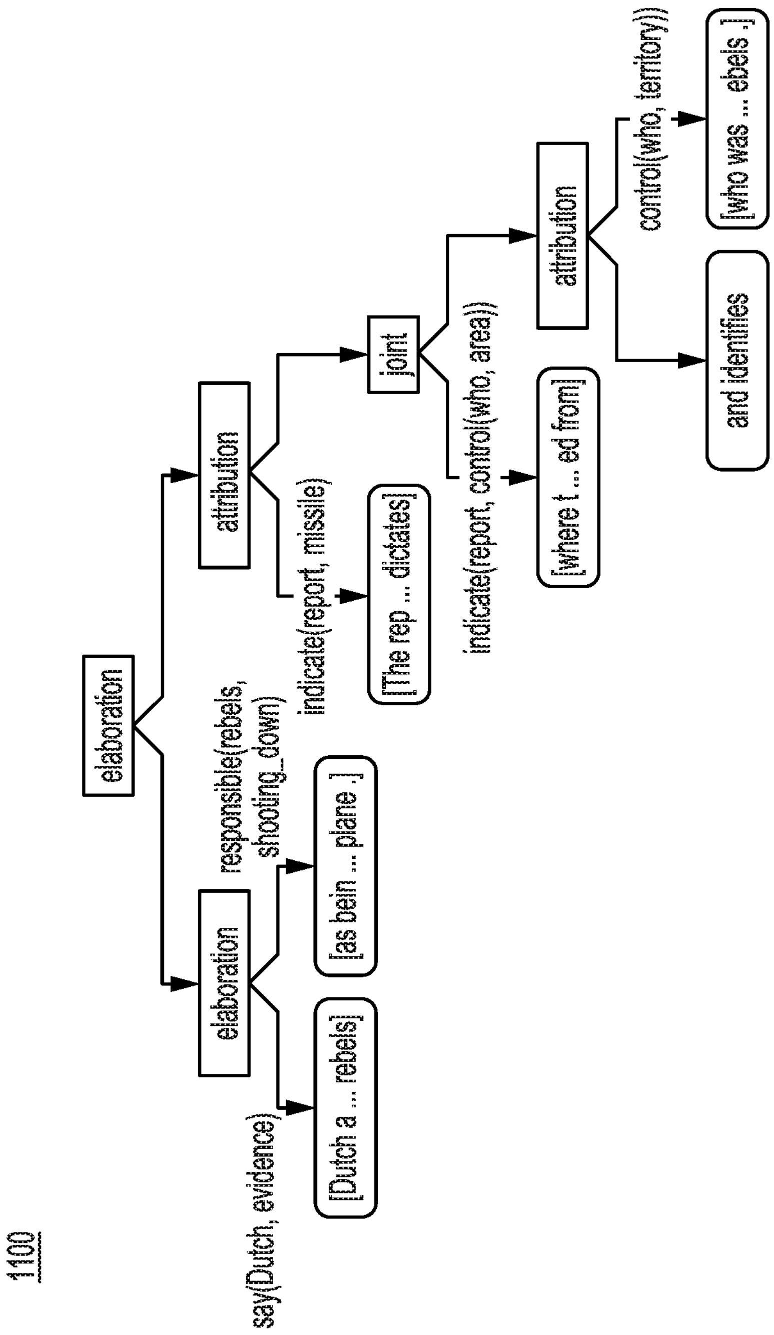

1100
elaboration
elaboration
attribution
say(Dutch, evidence)
[Dutch a ... rebels]
responsible(rebels, shooting_down)
[as bein ... plane .]
indicate(report, missile)
[The rep ... dictates]
joint
indicate(report, control(who, area))
[where t ... ed from]
attribution
and identifies
control(who, territory))
[who was ... ebels .]

FIG. 11

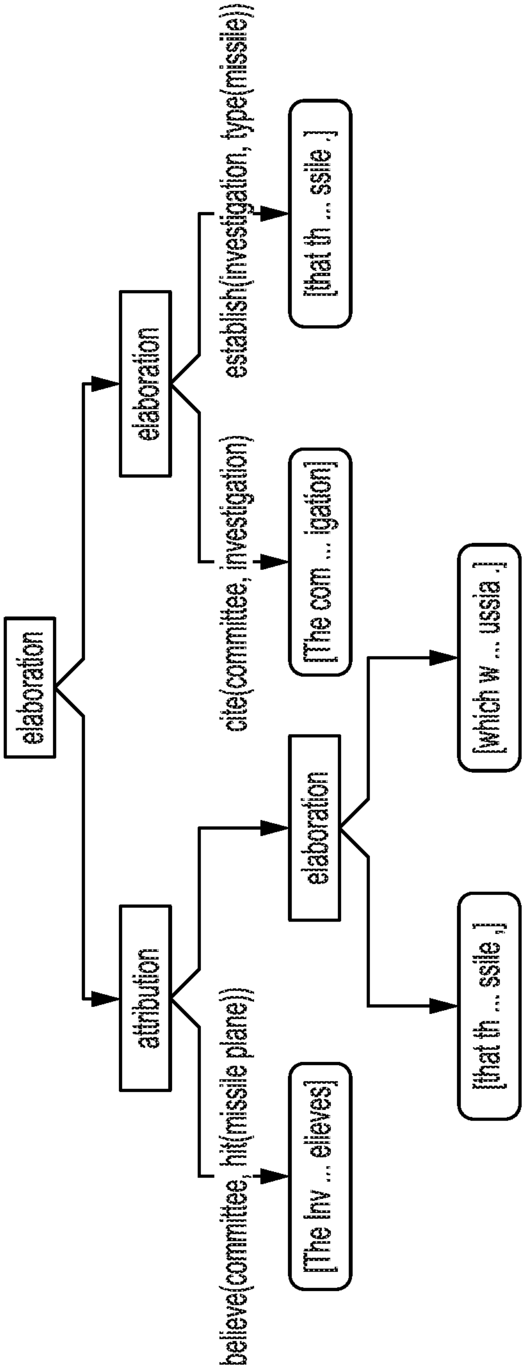

elaboration
attribution
elaboration
believe(committee, hit(missile plane))
[The Inv ... elieves]
elaboration
[that th ... ssile ,]
[which w ... ussia .]
cite(committee, investigation)
[The com ... igation]
establish(investigation, type(missile))
[that th ... ssile .]

FIG. 12

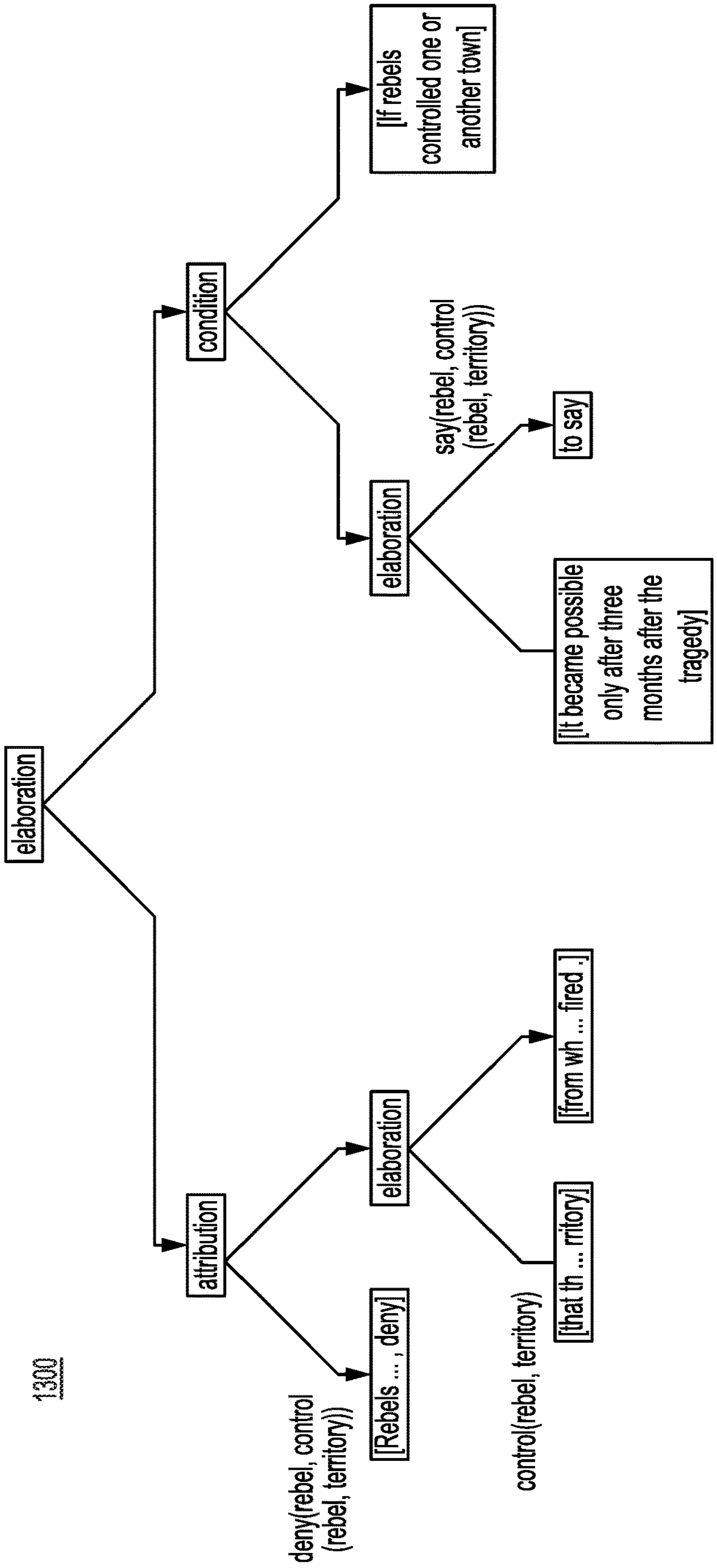

1300
elaboration
attribution
condition
deny(rebel, control (rebel, territory))
[Rebels ... , deny]
elaboration
control(rebel, territory)
[that th ... rritory]
[from wh ... fired .]
elaboration
say(rebel, control (rebel, territory))
[It became possible only after three months after the tragedy]
[to say
[If rebels controlled one or another town]

FIG. 13

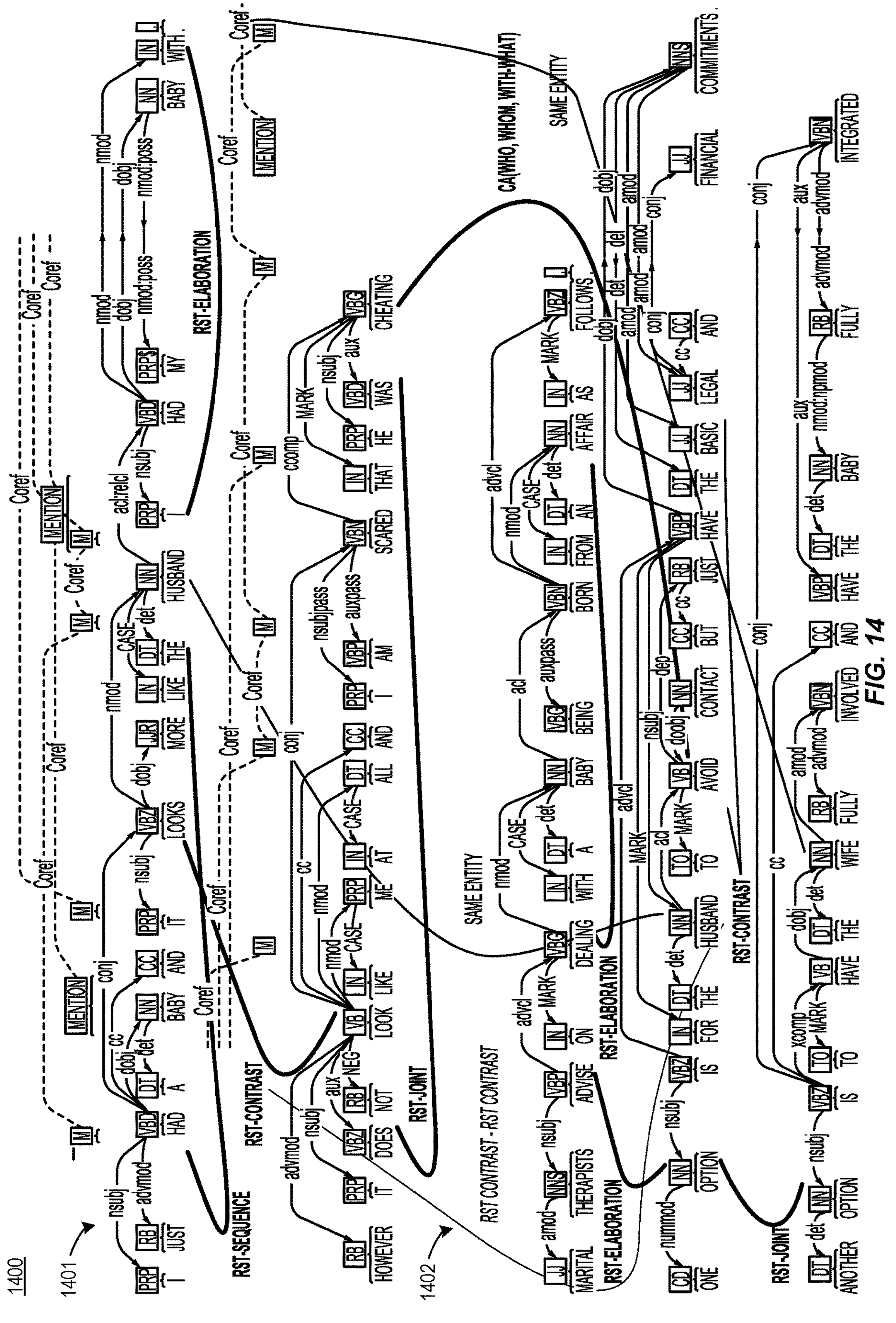

1400
1401
1402
I JUST HAD A BABY AND IT LOOKS MORE LIKE THE HUSBAND I HAD MY BABY WITH.
HOWEVER IT DOES NOT LOOK LIKE ME AT ALL AND I AM SCARED THAT HE WAS CHEATING
MARITAL THERAPISTS ADVISE ON DEALING WITH A BABY BEING BORN FROM AN AFFAIR AS FOLLOWS.
ONE OPTION IS FOR THE HUSBAND TO AVOID CONTACT BUT JUST HAVE THE BASIC LEGAL AND FINANCIAL COMMITMENTS.
ANOTHER OPTION IS TO HAVE THE WIFE FULLY INVOLVED AND HAVE THE BABY FULLY INTEGRATED
RST-SEQUENCE
RST-ELABORATION
RST-CONTRAST
RST-JOINT
RST CONTRAST - RST CONTRAST
SAME ENTITY
CA(WHO, WHOM, WITH-WHAT)
Coref
MENTION
M

FIG. 14

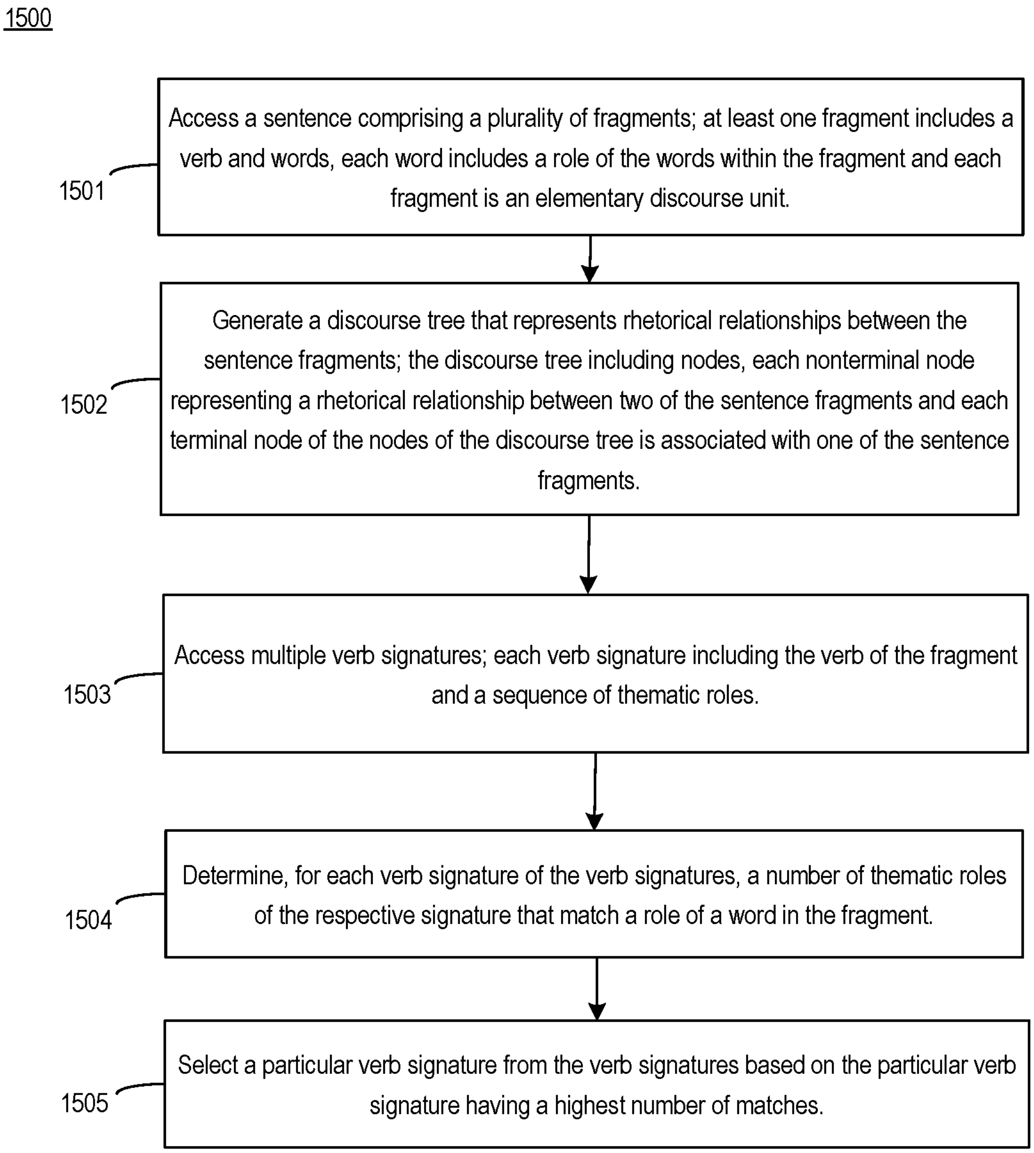

1500
1501
Access a sentence comprising a plurality of fragments; at least one fragment includes a verb and words, each word includes a role of the words within the fragment and each fragment is an elementary discourse unit.
1502
Generate a discourse tree that represents rhetorical relationships between the sentence fragments; the discourse tree including nodes, each nonterminal node representing a rhetorical relationship between two of the sentence fragments and each terminal node of the nodes of the discourse tree is associated with one of the sentence fragments.
1503
Access multiple verb signatures; each verb signature including the verb of the fragment and a sequence of thematic roles.
1504
Determine, for each verb signature of the verb signatures, a number of thematic roles of the respective signature that match a role of a word in the fragment.
1505
Select a particular verb signature from the verb signatures based on the particular verb signature having a highest number of matches.

FIG. 15

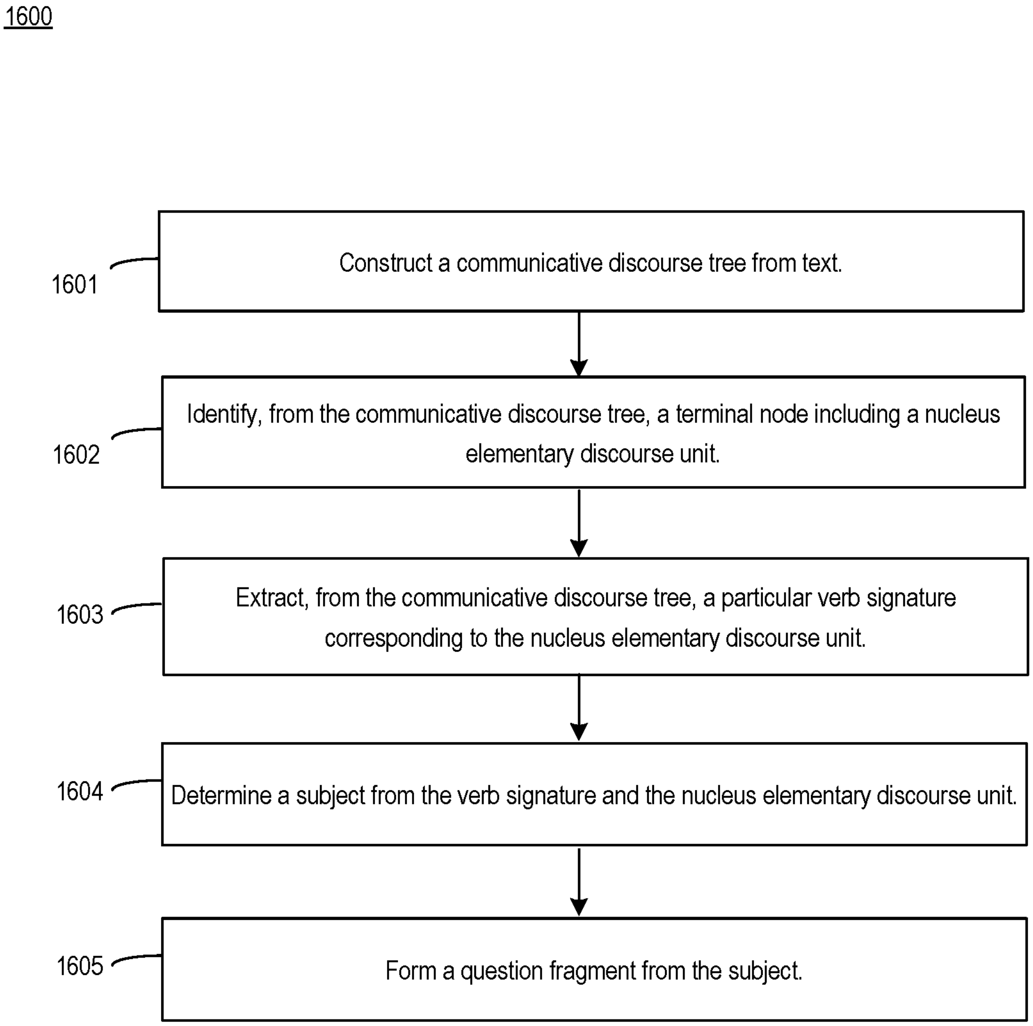

1600
1601
Construct a communicative discourse tree from text.
1602
Identify, from the communicative discourse tree, a terminal node including a nucleus elementary discourse unit.
1603
Extract, from the communicative discourse tree, a particular verb signature corresponding to the nucleus elementary discourse unit.
1604
Determine a subject from the verb signature and the nucleus elementary discourse unit.
1605
Form a question fragment from the subject.

FIG. 16

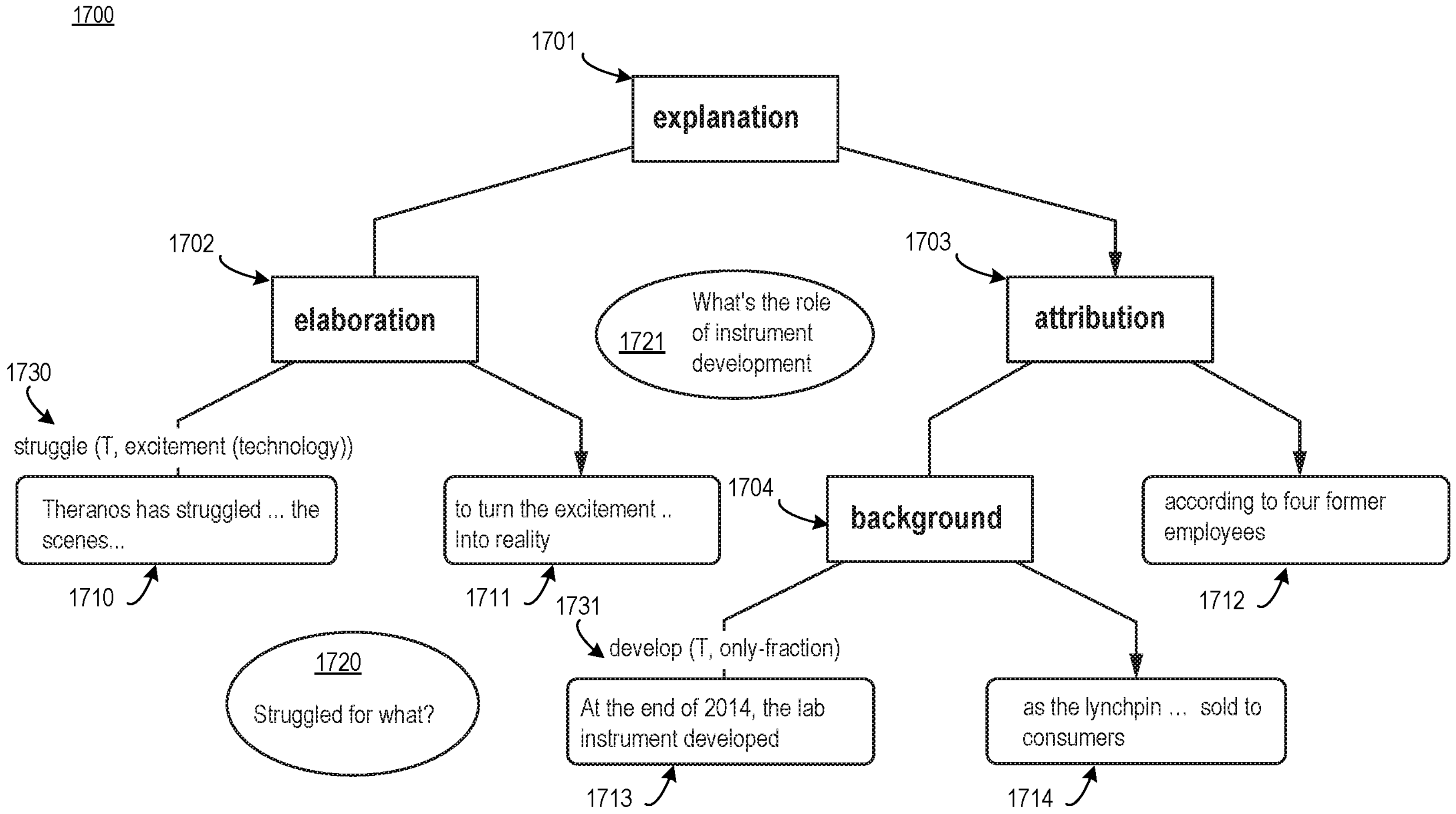

1700
1701
explanation
1702
elaboration
1703
attribution
1721
What's the role of instrument development
1730
struggle (T, excitement (technology))
Theranos has struggled ... the scenes...
1710
to turn the excitement .. Into reality
1711
1704
background
according to four former employees
1712
1731
develop (T, only-fraction)
1720
Struggled for what?
At the end of 2014, the lab instrument developed
1713
as the lynchpin ... sold to consumers
1714

FIG. 17

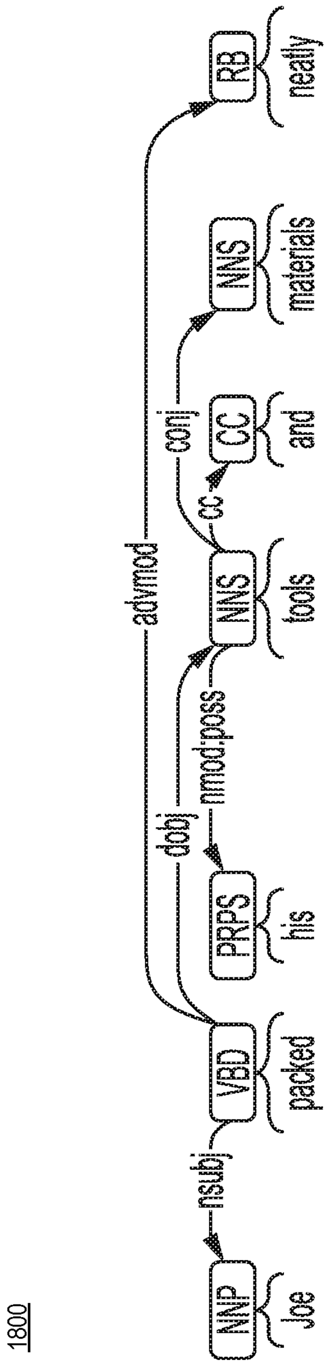

1800
NNP
Joe
nsubj
VBD
packed
PRPS
his
dobj
nmod:poss
advmod
NNS
tools
cc
CC
and
conj
NNS
materials
RB
neatly

FIG. 18

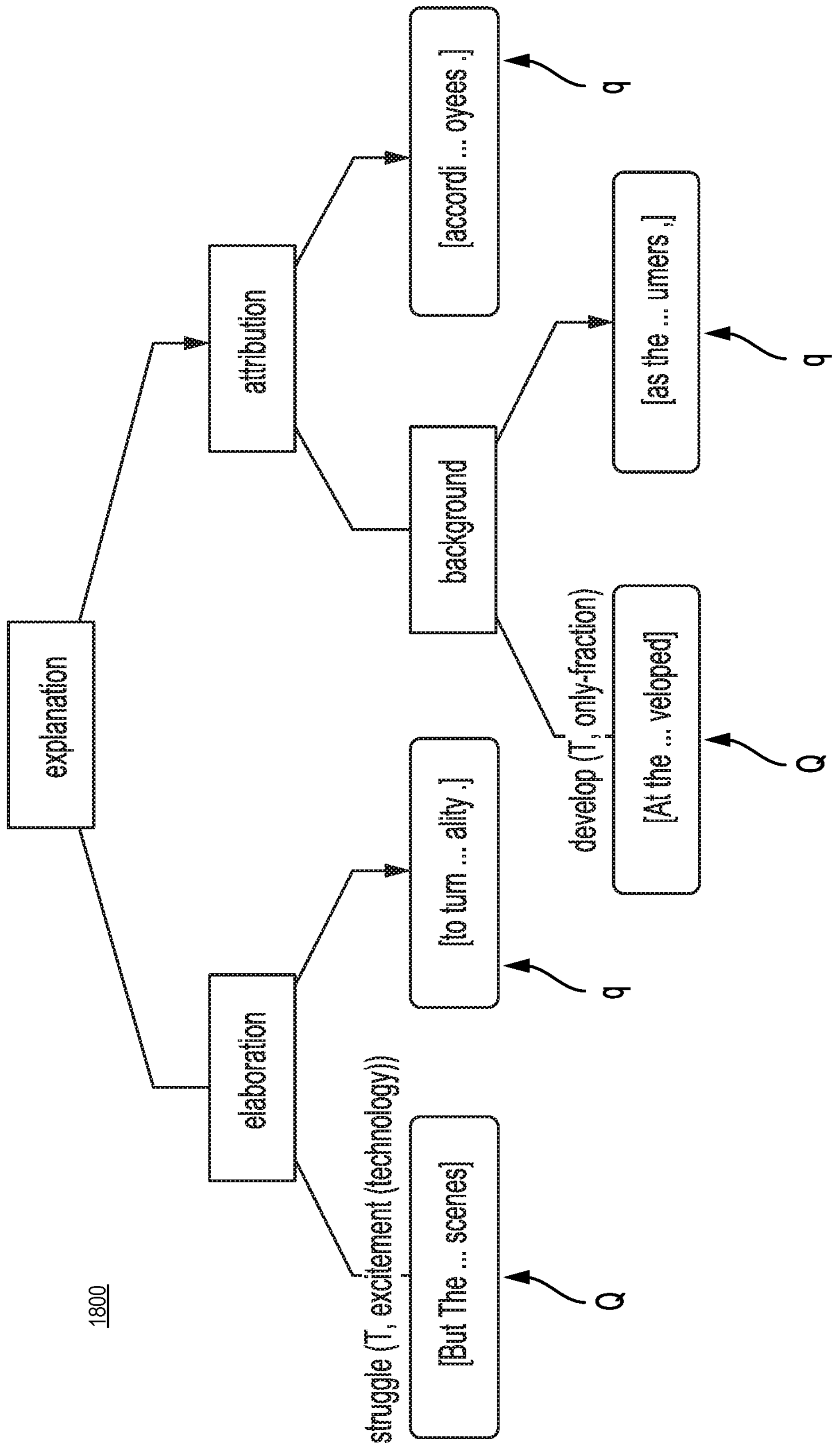

1800
explanation
elaboration
attribution
struggle (T, excitement (technology))
[But The ... scenes]
Q
[to turn ... ality .]
q
background
develop (T, only-fraction)
[At the ... veloped]
Q
[as the ... umers .]
q
[accordi ... oyees .]
q

FIG. 19

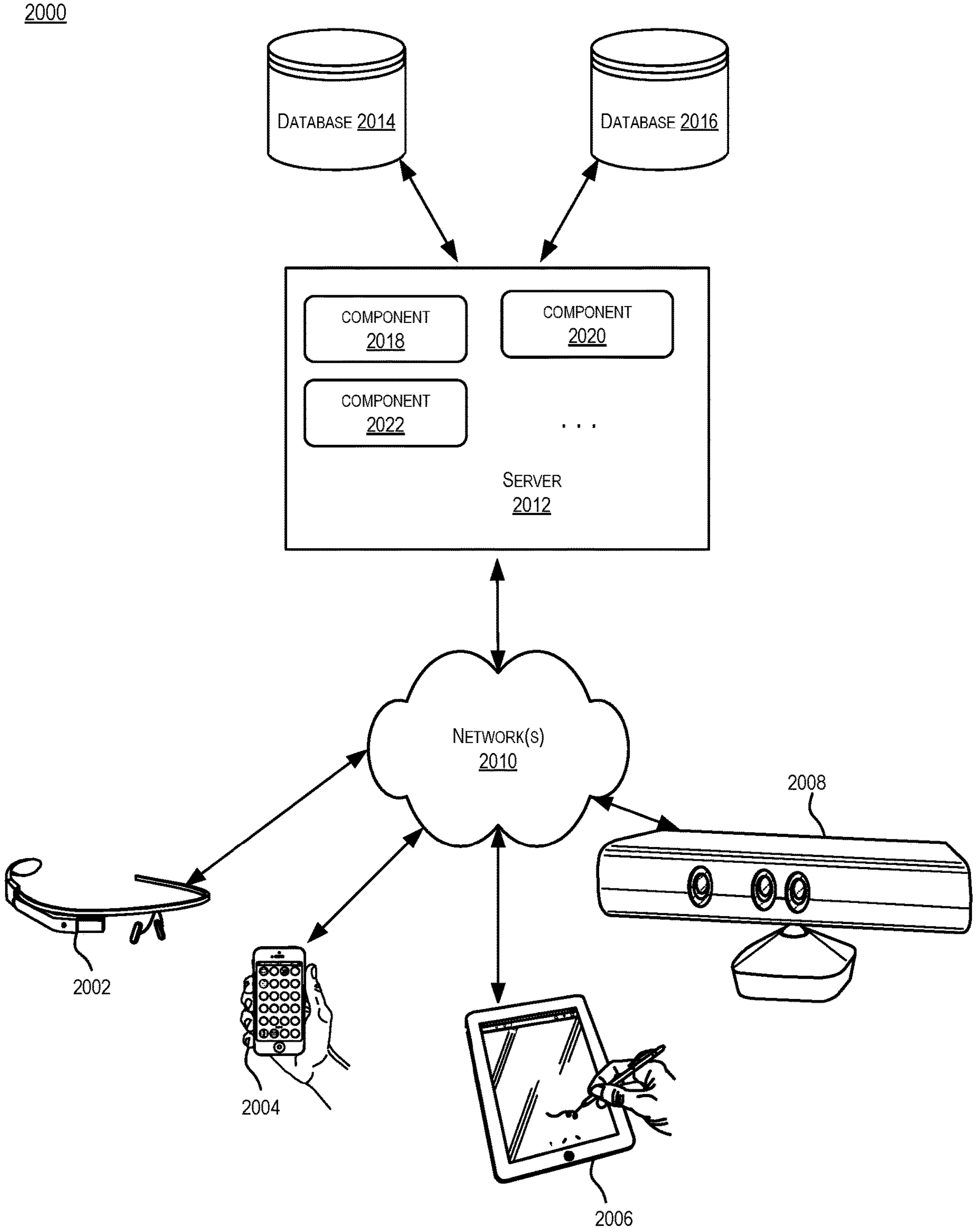

2000
Database 2014
Database 2016
Component 2018
Component 2020
Component 2022
. . .
Server 2012
Network(s) 2010
2008
2002
2004
2006

FIG. 20

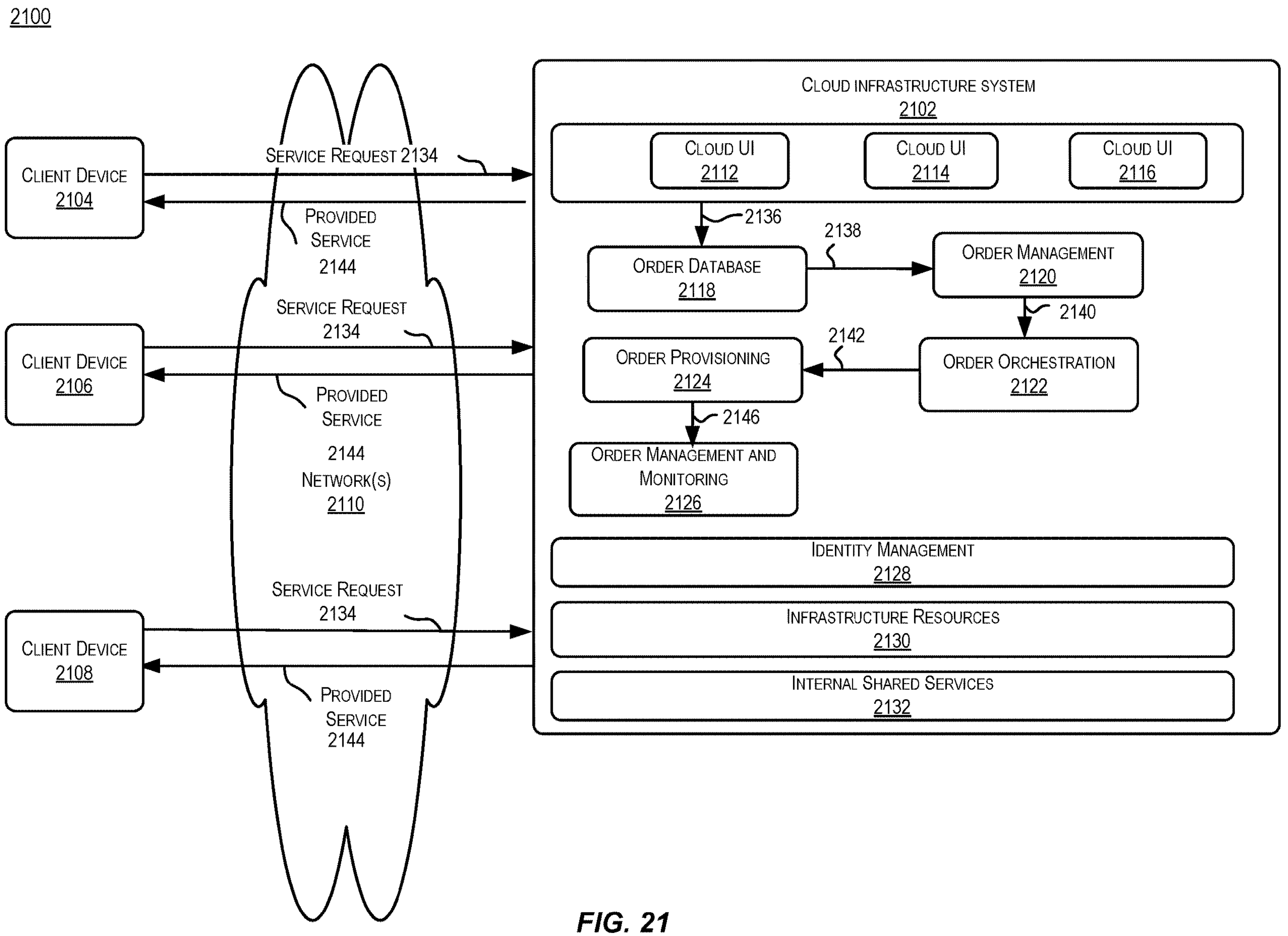

2100
Client Device 2104
Client Device 2106
Client Device 2108
Service Request 2134
Provided Service 2144
Service Request 2134
Provided Service 2144
Network(s) 2110
Service Request 2134
Provided Service 2144
Cloud infrastructure system 2102
Cloud UI 2112
Cloud UI 2114
Cloud UI 2116
2136
2138
Order Database 2118
Order Management 2120
2140
2142
Order Provisioning 2124
Order Orchestration 2122
2146
Order Management and Monitoring 2126
Identity Management 2128
Infrastructure Resources 2130
Internal Shared Services 2132

FIG. 21

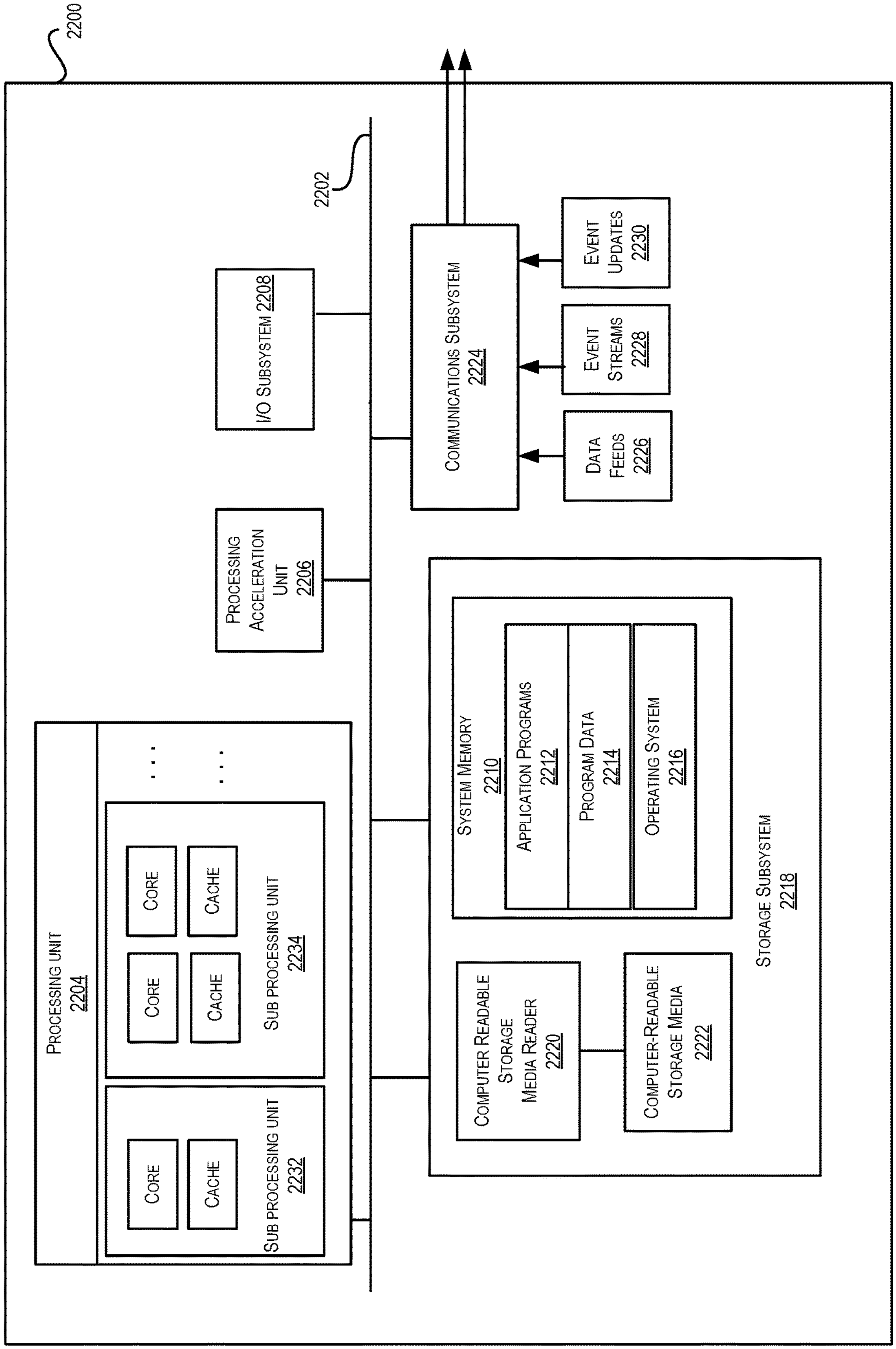

2200
2202
I/O Subsystem 2208
Processing Acceleration Unit 2206
Communications Subsystem 2224
Event Updates 2230
Event Streams 2228
Data Feeds 2226
Processing Unit 2204
Core
Cache
Core
Cache
Sub Processing Unit 2234
Core
Cache
Sub Processing Unit 2232
System Memory 2210
Application Programs 2212
Program Data 2214
Operating System 2216
Storage Subsystem 2218
Computer Readable Storage Media Reader 2220
Computer-Readable Storage Media 2222

FIG. 22

CONVERTING A DOCUMENT INTO A CHATBOT-ACCESSIBLE FORM VIA THE USE OF COMMUNICATIVE DISCOURSE TREES

CROSS-REFERENCES TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 62/894,162 filed Aug. 30, 2019 and is a continuation in part of U.S. application Ser. No. 16/789,849, filed Feb. 13, 2020, which claims the benefit of U.S. Provisional Application No. 62/832,986, filed Apr. 12, 2019 and is a continuation-in-part of U.S. application Ser. No. 15/975,683, filed May 9, 2018, which claims priority to U.S. Provisional Application No. 62/504,377, filed May 10, 2017. These applications are hereby incorporated by reference in their entireties for all purposes.

TECHNICAL FIELD

This disclosure is generally concerned with linguistics. More specifically, this disclosure relates to using communicative discourse trees to provide improved autonomous agents.

BACKGROUND

Computer-implemented applications of linguistics are increasing due to a greatly increased speed of processors and capacity of memory. For example, computer-based analysis of language discourse facilitates numerous applications, such as automated agents that can answer questions received from user devices.

Autonomous agents often leverage available electronic textual documents. But existing solutions are unable to fully leverage textual content. Hence, new solutions are needed.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows an exemplary textual analysis environment, in accordance with an aspect.

FIG. **2** depicts an example of a discourse tree in accordance with an aspect.

FIG. **3** depicts a further example of a discourse tree in accordance with an aspect.

FIG. **4** depicts illustrative schemas in accordance with an aspect.

FIG. **5** depicts a discourse tree in accordance with an aspect.

FIG. **6** depicts an exemplary indented text encoding of the representation in FIG. **5** in accordance with an aspect.

FIG. **7** depicts an exemplary discourse tree for an example request about property tax in accordance with an aspect.

FIG. **8** depicts a discourse tree for an exemplary response for the question represented in FIG. **7**.

FIG. **9** illustrates a discourse tree for an official answer in accordance with an aspect.

FIG. **10** illustrates a discourse tree for a raw answer in accordance with an aspect.

FIG. **11** illustrates a communicative discourse tree for a claim of a first agent in accordance with an aspect.

FIG. **12** illustrates a communicative discourse tree for a claim of a second agent in accordance with an aspect.

FIG. **13** illustrates a communicative discourse tree for a claim of a third agent in accordance with an aspect.

FIG. **14** illustrates parse thickets in accordance with an aspect.

FIG. **15** illustrates an exemplary process for building a communicative discourse tree in accordance with an aspect.

FIG. **16** illustrates an exemplary process for identifying questions from a text in accordance with an aspect.

FIG. **17** illustrates an exemplary communicative discourse tree in accordance with an aspect.

FIG. **18** illustrates an exemplary parse tree in accordance with an aspect.

FIG. **19** illustrates an exemplary communicative discourse tree that indicates global questions and dialogue questions for a text, in accordance with an aspect.

FIG. **20** depicts a simplified diagram of a distributed system for implementing one of the aspects.

FIG. **21** is a simplified block diagram of components of a system environment by which services provided by the components of an aspect system may be offered as cloud services in accordance with an aspect.

FIG. **22** illustrates an exemplary computer system, in which various aspects of the present disclosure may be implemented.

SUMMARY

Techniques are disclosed for generating a discourse structure from text. For example, disclosed techniques use discourse analysis to identify questions and answers from text. For example, the generated questions and answers can be used for facilitating improved autonomous agents, generating a virtual dialogue, generating training data for machine-learning models, or generating questions and answers from textual sources.

For example, an application implements a method. The method involves constructing, from text including fragments, a discourse tree that represents rhetorical relationships between the fragments. The discourse tree includes nodes. Each nonterminal node represents a rhetorical relationship between two of the fragments. Each terminal node of the nodes of the discourse tree is associated with one of the fragments. The fragments are elementary discourse units. The method further involves matching each fragment that has a verb to a verb signature, thereby creating a communicative discourse tree. The method further involves identifying, in the communicative discourse tree, a terminal node including a nucleus elementary discourse unit. The nucleus elementary discourse unit is dominant to a corresponding satellite elementary discourse unit. The method further involves extracting, from the communicative discourse tree, a particular verb signature corresponding to the nucleus elementary discourse unit. The method further involves determining a subject from the verb signature and the nucleus elementary discourse unit. The method further involves forming, from the subject, a question fragment.

In an aspect, forming the question fragment includes identifying, within the nucleus elementary discourse unit, a word that represents either (i) a noun, (ii) a verb, or (iii) adjective and replacing the word with a question word. A question word is one of (i) what, (ii) where, (iii) whom, (iv) who, or (v) how.

In an aspect, identifying a word includes constructing a parse tree from the nucleus elementary discourse unit. The parse tree includes nodes and selecting, from the nodes, a node that represents either (i) a noun, (ii) a verb, or (iii) adjective.

In an aspect, the question fragment includes accessing a first syntactic representation of a question template. Forming the question fragment further includes aligning a second syntactic representation of the question fragment with the first syntactic representation. Forming the question fragment further includes substituting the question fragment into the first syntactic representation.

In an aspect, the method further includes extracting a keyword from the question fragment. The method further includes submitting the keyword to a search engine. The method further includes obtaining a search result from the search engine. The method further includes comparing the search result with the question fragment by deriving a maximal common sub-tree of the question fragment and the search result. The method further includes responsive to determining that a number of words in the maximal common sub-tree is greater than a threshold, aligning a second syntactic representation of the question fragment with a first syntactic representation of the search result and substituting the question fragment into the second syntactic representation.

In an aspect, the method further includes determining the plurality of templates by performing operations. The operations include forming, from an additional text, a first parse tree from a first question and a second parse tree from a second question. The operations include identifying an entity from the first parse tree. The operations include identifying the entity from the second parse tree. The operations include generalizing the first parse tree and the second parse tree into a generalized fragment including the entity. The operations include adding the generalized fragment into the plurality of templates.

In an aspect, the text is in a first domain, and the method further includes generating, from an utterance, an additional communicative discourse tree. The method further includes applying a classification model to the additional communicative discourse tree. The classification model is trained with a plurality of questions and plurality of answers. The method further includes receiving, from the classification model, an indication of whether the utterance is in rhetoric agreement with a reference text. The reference text is in a second domain.

In an aspect, the method further includes deriving an answer from the nucleus elementary discourse unit. The method further includes training a classification model by inputting the question fragment and the answer to classification model. The method further includes providing the question fragment to the classification model. The method further includes verifying that the answer is received from the classification model.

The exemplary methods discussed above can be implemented on systems including one or more processors or stored as instructions on a non-transitory computer-readable medium.

DETAILED DESCRIPTION

Disclosed solutions provide technical improvements to the area of computer-implemented linguistics. More specifically, disclosed solutions use discourse analysis techniques to formulate questions and corresponding answers from textual sources. These questions and corresponding answers ("question-answer pairs") can be used to improve autonomous agents or "chatbots," for example by providing improved training of machine-learning models used by the agents. These improved autonomous agents therefore provide improvements relative to existing solutions, which may be capable of only scripted, or limited responses to user questions.

In particular, certain aspects can employ techniques such as rhetorical structure theory, communicative discourse trees, template matching, syntactic generalization, or web-mining. For example, rhetorical structure theory can be used to identify questions and answers from text. In another example, question templates can be applied to the generated questions, thereby verifying that a generated question is of a correct specificity. For example, a question should not be too specific as to give away the answer (e.g., "What is the name of a rock band from Liverpool, England with four members?").

As mentioned above, some aspects use "communicative discourse trees" or "CDTs." CDTs are discourse trees that are supplemented with one or more communicative actions. A communicative action is a cooperative action undertaken by individuals based on mutual deliberation and argumentation. Using communicative discourse trees and other techniques, improved performance over traditional statistical-based approaches can be accomplished. For example, communicative actions can be used to identify a subject of a portion of text. The identified subject can then be used to formulate an improved question.

Certain Definitions

As used herein, "rhetorical structure theory" is an area of research and study that provided a theoretical basis upon which the coherence of a discourse could be analyzed.

As used herein, "discourse tree" or "DT" refers to a structure that represents the rhetorical relations for a sentence of part of a sentence.

As used herein, a "rhetorical relation," "rhetorical relationship," or "coherence relation" or "discourse relation" refers to how two segments of discourse are logically connected to one another. Examples of rhetorical relations include elaboration, contrast, and attribution.

As used herein, a "sentence fragment," or "fragment" is a part of a sentence that can be divided from the rest of the sentence. A fragment is an elementary discourse unit. For example, for the sentence "Dutch accident investigators say that evidence points to pro-Russian rebels as being responsible for shooting down the plane," two fragments are "Dutch accident investigators say that evidence points to pro-Russian rebels" and "as being responsible for shooting down the plane." A fragment can, but need not, include a verb.

As used herein, "signature" or "frame" refers to a property of a verb in a fragment. Each signature can include one or more thematic roles. For example, for the fragment "Dutch accident investigators say that evidence points to pro-Russian rebels," the verb is "say" and the signature of this particular use of the verb "say" could be "agent verb topic" where "investigators" is the agent and "evidence" is the topic.

As used herein, "thematic role" refers to components of a signature used to describe a role of one or more words. Continuing the previous example, "agent" and "topic" are thematic roles.

As used herein, "nuclearity" refers to which text segment, fragment, or span, is more central to a writer's purpose. The nucleus is the more central span, and the satellite is the less central one.

As used herein, "coherency" refers to the linking together of two rhetorical relations.

As used herein, "communicative verb" is a verb that indicates communication. For example, the verb "deny" is a communicative verb.

As used herein, "communicative action" describes an action performed by one or more agents and the subjects of the agents.

Turning now to the Figures, FIG. **1** depicts an exemplary textual analysis environment in accordance with an aspect of the present disclosure. Textual analysis environment **100** includes one or more of input text **110**, computing device **101**, and user device **160**. In the example depicted in FIG. **1**, computing device **101** accesses input text **110**, generates questions **141** and **142** from input text **110**, and uses questions **141-142** to improve the performance of an autonomous agent. In another example depicted in FIG. **1**, computing device **101** implements an autonomous agent that interacts with user device **160**, as depicted in dialogue **162**.

Examples of input text **110** include electronic text source such as text files, Portable Document Format (PDF)® documents, rich text documents, and the like. In some cases, preprocessing may be performed on the input text **110** to remove unwanted characters or formatting fields. Input text **110** can be organized by using one or more structural or organizational approaches such as sections, paragraphs, pages, and the like.

Computing device **101** includes one or more of application **102**, text corpus **105**, classification model **120**, and training data **125**. Examples of computing device **101** are distributed system **2000** and client computing devices **2002**, **2004**, **2006**, and **2008**.

In some cases, application **102** can use machine learning techniques. For example, classification model **120**, which can be a predictive model or machine-learning model, can be trained to detect a presence of particular features within text or within a communicative discourse tree. Additional examples of classification models and approaches include support vector machines, nearest neighbor models, and tree kernel models.

In an example, classification model **120** can use one or more such models to analyze a communicative discourse tree generated from input text **110**. Examples of features that can be detected in text via the use of communicative discourse trees include a presence of argumentation, rhetoric agreement, a consecutive answer, or another specific feature present in input text **110**. Classification model **120** can be trained using and/or training data **125**, which can be generated by application **102** or another application or device.

In an aspect, application **102** can also populate text corpus **105** by generating questions and answers from input text **110**. For example, from input text **110**, application **102** generates questions, which can be stored in text corpus **105** and later used to answer questions from user device **160**. An example of a method for generating questions is discussed with respect to FIG. **16**.

User device **160** can be any mobile device such as a mobile phone, smart phone, tablet, laptop, smart watch, and the like. Application **102** can interact with user device **160**, for example, in a dialogue session. In a dialogue session, application **102** can receive questions from user device **160**, generate answers for the questions, and output the answers. Examples of user device **160** include client computing devices **2002**, **2004**, **2006**, and **2008**.

In an aspect, application **102** can facilitate a dialogue **162** with user device **160** by using questions and answers from text corpus **105**. For example, application **102** generates questions and answers from one or more corpuses of text such as text corpus **105**, which can be local to computing device **101**. In an aspect, the generation of content can involve creating one or more communicative discourse trees from questions received from user device **160** and use the generated communicative discourse trees to generate an answer to the question. In an aspect, application **102** can annotate questions and answers within the output dialogue **150** as appearing to be from an autonomous agent or a user. For example "User1: struggled for what?" and "Agent2: To turn excitement into reality."

Rhetoric Structure Theory and Discourse Trees

Linguistics is the scientific study of language. For example, linguistics can include the structure of a sentence (syntax), e.g., subject-verb-object, the meaning of a sentence (semantics), e.g. dog bites man vs. man bites dog, and what speakers do in conversation, i.e., discourse analysis or the analysis of language beyond the sentence.

The theoretical underpinnings of discourse, Rhetoric Structure Theory (RST), can be attributed to Mann, William and Thompson, Sandra, "Rhetorical structure theory: A Theory of Text organization," Text-Interdisciplinary Journal for the Study of Discourse, 8(3):243-281, 1988. Similar to how the syntax and semantics of programming language theory helped enable modern software compilers, RST helped enabled the analysis of discourse. More specifically RST posits structural blocks on at least two levels, a first level such as nuclearity and rhetorical relations, and a second level of structures or schemas. Discourse parsers or other computer software can parse text into a discourse tree.

Rhetoric Structure Theory models logical organization of text, a structure employed by a writer, relying on relations between parts of text. RST simulates text coherence by forming a hierarchical, connected structure of texts via discourse trees. Rhetoric relations are split into the classes of coordinate and subordinate; these relations hold across two or more text spans and therefore implement coherence. These text spans are called elementary discourse units (EDUs). Clauses in a sentence and sentences in a text are logically connected by the author. The meaning of a given sentence is related to that of the previous and the following sentences. This logical relation between clauses is called the coherence structure of the text. RST is one of the most popular theories of discourse, being based on a tree-like discourse structure, discourse trees (DTs). The leaves of a DT correspond to EDUs, the contiguous atomic text spans. Adjacent EDUs are connected by coherence relations (e.g., Attribution, Sequence), forming higher-level discourse units. These units are then also subject to this relation linking. EDUs linked by a relation are then differentiated based on their relative importance: nuclei are the core parts of the relation, while satellites are peripheral ones. As discussed, in order to determine accurate request-response pairs, both topic and rhetorical agreement are analyzed. When a speaker answers a question, such as a phrase or a sentence, the speaker's answer should address the topic of this question. In the case of an implicit formulation of a question, via a seed text of a message, an appropriate answer is expected not only maintain a topic, but also match the generalized epistemic state of this seed.

Rhetoric Relations

As discussed, aspects described herein use communicative discourse trees. Rhetorical relations can be described in different ways. For example, Mann and Thompson describe twenty-three possible relations. C. Mann, William & Thompson, Sandra. (1987) ("Mann and Thompson"). Rhetorical Structure Theory: A Theory of Text Organization. Other numbers of relations are possible.

TABLE 1

Relation Name	Nucleus	Satellite
Antithesis	ideas favored by the author	ideas disfavored by the author
Background	text whose understanding is being facilitated	text for facilitating understanding
Circumstance	text expressing the events or ideas occurring in the interpretive context	an interpretive context of situation or time
Concession	situation affirmed by author	situation which is apparently inconsistent but also affirmed by author
Condition	action or situation whose occurrence results from the occurrence of the conditioning situation	conditioning situation
Elaboration	basic information	additional information
Enablement	an action	information intended to aid the reader in performing an action
Evaluation	a situation	an evaluative comment about the situation
Evidence	a claim	information intended to increase the reader's belief in the claim
Interpretation	a situation	an interpretation of the situation
Justify	text	information supporting the writer's right to express the text
Motivation	an action	information intended to increase the reader's desire to perform the action
Non-volitional Cause	a situation	another situation which causes that one, but not by anyone's deliberate action
Non-volitional Result	a situation	another situation which is caused by that one, but not by anyone's deliberate action
Otherwise (anti conditional)	action or situation whose occurrence results from the lack of occurrence of the conditioning situation	conditioning situation
Purpose	an intended situation	the intent behind the situation
Restatement	a situation	a reexpression of the situation
Solutionhood	a situation or method supporting full or partial satisfaction of the need	a question, request, problem, or other expressed need
Summary	text	a short summary of that text
Volitional Cause	a situation	another situation which causes that one, by someone's deliberate action
Volitional Result	a situation	another situation which is caused by that one, by someone's deliberate action

Some empirical studies postulate that the majority of text is structured using nucleus-satellite relations. See Mann and Thompson. But other relations do not carry a definite selection of a nucleus. Examples of such relations are shown below.

TABLE 2

Relation Name	Span	Other Span
Contrast	One alternate	The other alternate
Joint	(unconstrained)	(unconstrained)
List	An item	A next item
Sequence	An item	A next item

FIG. **2** depicts an example of a discourse tree, in accordance with an aspect. FIG. **2** includes discourse tree **200**. Discourse tree includes text span **201**, text span **202**, text span **203**, relation **210** and relation **228**. The numbers in FIG. **2** correspond to the three text spans. FIG. **3** corresponds to the following example text with three text spans numbered 1, 2, 3:

1. Honolulu, Hi. will be site of the 2016 Conference on Hawaiian History

2. It is expected that **200** historians from the U.S. and Asia will attend

3. The conference will be concerned with how the Polynesians sailed to Hawaii

For example, relation **210**, or elaboration, describes the relationship between text span **201** and text span **202**. Relation **228** depicts the relationship, elaboration, between text span **203** and **204**. As depicted, text spans **202** and **203** elaborate further on text span **201**. In the above example, given a goal of notifying readers of a conference, text span 1 is the nucleus. Text spans 2 and 3 provide more detail about the conference. In FIG. **2**, a horizontal number, e.g., 1-3, 1, 2, 3 covers a span of text (possibly made up of further spans); a vertical line signals the nucleus or nuclei; and a curve represents a rhetoric relation (elaboration) and the direction of the arrow points from the satellite to the nucleus.

If the text span only functions as a satellite and not as a nuclei, then deleting the satellite would still leave a coherent text. If from FIG. **2** one deletes the nucleus, then text spans 2 and 3 are difficult to understand.

FIG. **3** depicts a further example of a discourse tree in accordance with an aspect. FIG. **3** includes components **301** and **302**, text spans **305**-**307**, relation **310** and relation **328**. Relation **310** depicts the relationship **310**, enablement, between components **306** and **305**, and **307**, and **305**. FIG. **3** refers to the following text spans:

1. The new Tech Report abstracts are now in the journal area of the library near the abridged dictionary.

2. Please sign your name by any means that you would be interested in seeing.

3. Last day for sign-ups is 31 May.

As can be seen, relation **328** depicts the relationship between entity **307** and **306**, which is enablement. FIG. **3** illustrates that while nuclei can be nested, there exists only one most nuclear text span.

Constructing a Discourse Tree

Discourse trees can be generated using different methods. A simple example of a method to construct a DT bottom up is:

(1) Divide the discourse text into units by:

(a) Unit size may vary, depending on the goals of the analysis

(b) Typically, units are clauses

(2) Examine each unit, and its neighbors. Is there a relation holding between them?

(3) If yes, then mark that relation.

(4) If not, the unit might be at the boundary of a higher-level relation. Look at relations holding between larger units (spans).

(5) Continue until all the units in the text are accounted for.

Mann and Thompson also describe the second level of building block structures called schemas applications. In RST, rhetoric relations are not mapped directly onto texts; they are fitted onto structures called schema applications, and these in turn are fitted to text. Schema applications are derived from simpler structures called schemas (as shown by FIG. **4**). Each schema indicates how a particular unit of text is decomposed into other smaller text units. A rhetorical structure tree or DT is a hierarchical system of schema applications. A schema application links a number of consecutive text spans, and creates a complex text span, which can in turn be linked by a higher-level schema application. RST asserts that the structure of every coherent discourse can be described by a single rhetorical structure tree, whose top schema creates a span encompassing the whole discourse.

FIG. **4** depicts illustrative schemas, in accordance with an aspect. FIG. **4** shows a joint schema is a list of items consisting of nuclei with no satellites. FIG. **4** depicts schemas **401**-**406**. Schema **401** depicts a circumstance relation between text spans **410** and **428**. Scheme **402** depicts a sequence relation between text spans **420** and **421** and a sequence relation between text spans **421** and **422**. Schema **403** depicts a contrast relation between text spans **430** and **431**. Schema **404** depicts a joint relationship between text spans **440** and **441**. Schema **405** depicts a motivation relationship between **450** and **451**, and an enablement relationship between **452** and **451**. Schema **406** depicts joint relationship between text spans **460** and **462**. An example of a joint scheme is shown in FIG. **4** for the three text spans below:

1. Skies will be partly sunny in the New York metropolitan area today.

2. It will be more humid, with temperatures in the middle **80**'s.

3. Tonight will be mostly cloudy, with the low temperature between **65** and **70**.

While FIGS. **2**-**4** depict some graphical representations of a discourse tree, other representations are possible.

FIG. **5** depicts a discourse tree in accordance with an aspect. As can be seen from FIG. **5**, the leaves of a DT correspond to contiguous non-overlapping text spans called Elementary Discourse Units (EDUs). Adjacent EDUs are connected by relations (e.g., elaboration, attribution . . .) and form larger discourse units, which are also connected by relations. "Discourse analysis in RST involves two subtasks: discourse segmentation is the task of identifying the EDUs, and discourse parsing is the task of linking the discourse units into a labeled tree." See Joty, Shafiq R and Giuseppe Carenini, Raymond T Ng, and Yashar Mehdad. 2013. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In ACL (1), pages 486-496.

FIG. **5** depicts text spans that are leaves, or terminal nodes, on the tree, each numbered in the order they appear in the full text, shown in FIG. **6**. FIG. **5** includes tree **500**. Tree **500** includes, for example, nodes **501**-**507**. The nodes indicate relationships. Nodes are non-terminal, such as node **501**, or terminal, such as nodes **502**-**507**. As can be seen, nodes **503** and **504** are related by a joint relationship. Nodes **502**, **505**, **506**, and **508** are nuclei. The dotted lines indicate that the branch or text span is a satellite. The relations are nodes in gray boxes.

FIG. **6** depicts an exemplary indented text encoding of the representation in FIG. **5** in accordance with an aspect. FIG. **6** includes text **600** and text sequences **602**-**604**. Text **600** is presented in a manner more amenable to computer programming. Text sequence **602** corresponds to node **502**, sequence **603** corresponds to node **503**, and sequence **604** corresponds to node **504**. In FIG. **6**, "N" indicates a nucleus and "S" indicates a satellite.

Examples of Discourse Parsers

Automatic discourse segmentation can be performed with different methods. For example, given a sentence, a segmentation model identifies the boundaries of the composite elementary discourse units by predicting whether a boundary should be inserted before each particular token in the sentence. For example, one framework considers each token in the sentence sequentially and independently. In this framework, the segmentation model scans the sentence token by token, and uses a binary classifier, such as a support vector machine or logistic regression, to predict whether it is appropriate to insert a boundary before the token being examined. In another example, the task is a sequential labeling problem. Once text is segmented into elementary discourse units, sentence-level discourse parsing can be performed to construct the discourse tree. Machine learning techniques can be used.

In one aspect of the present disclosure, two Rhetorical Structure Theory (RST) discourse parsers are used: CoreNLPProcessor which relies on constituent syntax, and FastNLPProcessor which uses dependency syntax. See Surdeanu, Mihai & Hicks, Thomas & Antonio Valenzuela-Escarcega, Marco. Two Practical Rhetorical Structure Theory Parsers. (2015).

In addition, the above two discourse parsers, i.e., CoreNLPProcessor and FastNLPProcessor use Natural Language Processing (NLP) for syntactic parsing. For example, the Stanford CoreNLP gives the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and syntactic dependencies, indicate which noun phrases refer to the same entities. Practically, RST is a still theory that may work in many cases of discourse, but in some cases, it may not work. There are many variables including, but not limited to, what EDU's are in a coherent text, i.e., what discourse segmenters are used, what relations inventory is used and what relations are selected for the EDUs, the corpus of documents used for training and testing, and even what parsers are used. So for example, in Surdeanu, et al., "Two Practical Rhetorical Structure Theory Parsers," paper cited above, tests must be run on a particular corpus using specialized metrics to determine which parser gives better performance. Thus unlike computer language parsers which give predictable results, discourse parsers (and segmenters) can give unpredictable results depending on the training and/or test text corpus. Thus, discourse trees are a mixture of the predicable arts (e.g., compilers) and the unpredictable arts (e.g., like chemistry were experimentation is needed to determine what combinations will give you the desired results).

In order to objectively determine how good a Discourse analysis is, a series of metrics are being used, e.g., Precision/Recall/F1 metrics from Daniel Marcu, "The Theory and Practice of Discourse Parsing and Summarization," MIT Press, (2000). Precision, or positive predictive value is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Both precision and recall are therefore based on an understanding and measure of relevance. Suppose a computer program for recognizing dogs in photographs identifies eight dogs in a picture containing 12 dogs and some cats. Of the eight dogs identified, five actually are dogs (true positives), while the rest are cats (false positives). The program's precision is 5/8 while its recall is 5/12. When a search engine returns 30 pages only 20 of which were relevant while failing to return 40 additional relevant pages, its precision is 20/30=2/3 while its recall is 20/60=1/3. Therefore, in this case, precision is 'how useful the search results are', and recall is 'how complete the results are.'" The F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision and the recall of the test to compute the score: F1=2×((precision×recall)/(precision+recall)) and is the harmonic mean of precision and recall. The F1 score reaches its best value at 1 (perfect precision and recall) and worst at **0**.

Autonomous Agents or Chatbots

A conversation between Human A and Human B is a form of discourse. For example, applications exist such as FaceBook® Messenger, WhatsApp®, Slack,® SMS, etc., a conversation between A and B may typically be via messages in addition to more traditional email and voice conversations. A chatbot (which may also be called intelligent bots or virtual assistant, etc.) is an "intelligent" machine that, for example, replaces human B and to various degrees mimics the conversation between two humans. An example ultimate goal is that human A cannot tell whether B is a human or a machine (the Turning test, developed by Alan Turing in 1950). Discourse analysis, artificial intelligence, including machine learning, and natural language processing, have made great strides toward the long-term goal of passing the Turing test. Of course, with computers being more and more capable of searching and processing vast repositories of data and performing complex analysis on the data to include predictive analysis, the long-term goal is the chatbot being human-like and a computer combined.

For example, users can interact with the Intelligent Bots Platform through a conversational interaction. This interaction, also called the conversational user interface (UI), is a dialog between the end user and the chatbot, just as between two human beings. It could be as simple as the end user saying "Hello" to the chatbot and the chatbot responding with a "Hi" and asking the user how it can help, or it could be a transactional interaction in a banking chatbot, such as transferring money from one account to the other, or an informational interaction in a HR chatbot, such as checking for vacation balance, or asking an FAQ in a retail chatbot, such as how to handle returns. Natural language processing (NLP) and machine learning (ML) algorithms combined with other approaches can be used to classify end user intent. An intent at a high level is what the end user would like to accomplish (e.g., get account balance, make a purchase). An intent is essentially, a mapping of customer input to a unit of work that the backend should perform. Therefore, based on the phrases uttered by the user in the chatbot, these are mapped that to a specific and discrete use case or unit of work, for e.g. check balance, transfer money and track spending are all "use cases" that the chatbot should support and be able to work out which unit of work should be triggered from the free text entry that the end user types in a natural language.

The underlying rational for having an AI chatbot respond like a human is that the human brain can formulate and understand the request and then give a good response to the human request much better than a machine. Thus, there should be significant improvement in the request/response of a chatbot, if human B is mimicked. So an initial part of the problem is how does the human brain formulate and understand the request? To mimic, a model is used. RST and DT allow a formal and repeatable way of doing this.

At a high level, there are typically two types of requests: (1) A request to perform some action; and (2) a request for information, e.g., a question. The first type has a response in which a unit of work is created. The second type has a response that is, e.g., a good answer, to the question. The answer could take the form of, for example, in some aspects, the AI constructing an answer from its extensive knowledge base(s) or from matching the best existing answer from searching the internet or intranet or other publically/privately available data sources.

Communicative Discourse Trees and the Rhetoric Classifier

Aspects of the present disclosure build communicative discourse trees and use communicative discourse trees to analyze whether the rhetorical structure of a request or question agrees with an answer. More specifically, aspects described herein create representations of a request-response pair, learns the representations, and relates the pairs into classes of valid or invalid pairs. In this manner, an autonomous agent can receive a question from a user, process the question, for example, by searching for multiple answers, determine the best answer from the answers, and provide the answer to the user.

More specifically, to represent linguistic features of text, aspects described herein use rhetoric relations and speech acts (or communicative actions). Rhetoric relations are relationships between the parts of the sentences, typically obtained from a discourse tree. Speech acts are obtained as verbs from a verb resource such as VerbNet. By using both rhetoric relations and communicative actions, aspects described herein can correctly recognize valid request-response pairs. To do so, aspects correlate the syntactic structure of a question with that of an answer. By using the structure, a better answer can be determined.

For example, when an autonomous agent receives an indication from a person that the person desires to sell an item with certain features, the autonomous agent should provide a search result that not only contains the features but also indicates an intent to buy. In this manner, the autonomous agent has determined the user's intent. Similarly, when an autonomous agent receives a request from a person to share knowledge about a particular item, the search result should contain an intent to receive a recommendation. When a person asks an autonomous agent for an opinion about a subject, the autonomous agent shares an opinion about the subject, rather than soliciting another opinion.

Analyzing Request and Response Pairs

FIG. **7** depicts an exemplary discourse tree for an example request about property tax in accordance with an aspect. The node labels are the relations and the arrowed line points to the satellite. The nucleus is a solid line. FIG. **7** depicts the following text.

Request: "My husbands' grandmother gave him his grandfather's truck. She signed the title over but due to my husband having unpaid fines on his license, he was not able to get the truck put in his name. I wanted to put in my name and paid the property tax and got insurance for the truck. By the time it came to sending off the title and getting the tag, I didn't have the money to do so. Now, due to circumstances, I am not going to be able to afford the truck. I went to the insurance place and was refused a refund. I am just wondering that since I am not going to have a tag on this truck, is it possible to get the property tax refunded?"

Response: "The property tax is assessed on property that you own. Just because you chose to not register it does not mean that you don't own it, so the tax is not refundable. Even if you have not titled the vehicle yet, you still own it within the boundaries of the tax district, so the tax is payable. Note that all states give you a limited amount of time to transfer title and pay the use tax. If you apply late, there will be penalties on top of the normal taxes and fees. You don't need to register it at the same time, but you absolutely need to title it within the period of time stipulated in state law."

As can be seen in FIG. **7**, analyzing the above text results in the following. "My husbands' grandmother gave him his grandfather's truck" is elaborated by "She signed the title over but due to my husband" elaborated by "having unpaid fines on his license, he was not able to get the truck put in his name." which is elaborated by "I wanted to put in my name," "and paid the property tax", and "and got insurance for the truck."

"My husbands' grandmother gave him his grandfather's truck. She signed the title over but due to my husband having unpaid fines on his license, he was not able to get the truck put in his name. I wanted to put in my name and paid the property tax and got insurance for the truck." is elaborated by;

"I didn't have the money" elaborated by "to do so" contrasted with [**0107**] "By the time" elaborated by "it came to sending off the title"

"and getting the tag"

"My husbands' grandmother gave him his grandfather's truck. She signed the title over but due to my husband having unpaid fines on his license, he was not able to get the truck put in his name. I wanted to put in my name and paid the property tax and got insurance for the truck. By the time it came to sending off the title and getting the tag, I didn't have the money to do so" is contrasted with

"Now, due to circumstances," elaborated with "I am not going to be able to afford the truck." which is elaborated with

"I went to the insurance place"

"and was refused a refund"

"My husbands' grandmother gave him his grandfather's truck. She signed the title over but due to my husband having unpaid fines on his license, he was not able to get the truck put in his name. I wanted to put in my name and paid the property tax and got insurance for the truck. By the time it came to sending off the title and getting the tag, I didn't have the money to do so. Now, due to circumstances, I am not going to be able to afford the truck. I went to the insurance place and was refused a refund." is elaborated with

"I am just wondering that since I am not going to have a tag on this truck, is it possible to get the property tax refunded?"

"I am just wondering" has attribution to

"that" is the same unit as "is it possible to get the property tax refunded?" which has condition "since I am not going to have a tag on this truck"

As can be seen, the main subject of the topic is "Property tax on a car". The question includes the contradiction: on one hand, all properties are taxable, and on the other hand, the ownership is somewhat incomplete. A good response has to address both topic of the question and clarify the inconsistency. To do that, the responder is making even stronger claim concerning the necessity to pay tax on whatever is owned irrespectively of the registration status. This example is a member of positive training set from our Yahoo! Answers evaluation domain. The main subject of the topic is "Property tax on a car". The question includes the contradiction: on one hand, all properties are taxable, and on the other hand, the ownership is somewhat incomplete. A good answer/response has to address both topic of the question and clarify the inconsistency. The reader can observe that since the question includes rhetoric relation of contrast, the answer has to match it with a similar relation to be convincing. Otherwise, this answer would look incomplete even to those who are not domain experts.

FIG. **8** depicts a discourse tree for an exemplary response for the question represented in FIG. **7**, according to certain aspects of the present disclosure. The central nucleus is "the property tax is assessed on property" elaborated by "that you own". "The property tax is assessed on property that you own" is also a nucleus elaborated by "Just because you chose to not register it does not mean that you don't own it, so the tax is not refundable. Even if you have not titled the vehicle yet, you still own it within the boundaries of the tax district, so the tax is payable. Note that all states give you a limited amount of time to transfer title and pay the use tax."

The nucleus "The property tax is assessed on property that you own. Just because you chose to not register it does not mean that you don't own it, so the tax is not refundable. Even if you have not titled the vehicle yet, you still own it within the boundaries of the tax district, so the tax is payable. Note that all states give you a limited amount of time to transfer title and pay the use tax." is elaborated by "there will be penalties on top of the normal taxes and fees" with condition "If you apply late," which in turn is elaborated by the contrast of "but you absolutely need to title it within the period of time stipulated in state law." and "You don't need to register it at the same time.".

Comparing the DT of FIG. **7** and DT of FIG. **8**, enables a determination of how well matched the response (FIG. **8**) is to the request (FIG. **7**). In some aspects of the present disclosure, the above framework is used, at least in part, to determine the DTs for the request/response and the rhetoric agreement between the DTs.

In another example, the question "What does The Investigative Committee of the Russian Federation do" has at least two answers, for example, an official answer or an actual answer.

FIG. **9** illustrates a discourse tree for an official answer in accordance with an aspect. As depicted in FIG. **9**, an official answer, or mission statement states that "The Investigative Committee of the Russian Federation is the main federal investigating authority which operates as Russia's Anti-corruption agency and has statutory responsibility for inspecting the police forces, combating police corruption and police misconduct, is responsible for conducting investigations into local authorities and federal governmental bodies."

FIG. **10** illustrates a discourse tree for a raw answer in accordance with an aspect. As depicted in FIG. **10**, another, perhaps more honest, answer states that "Investigative Committee of the Russian Federation is supposed to fight corruption. However, top-rank officers of the Investigative Committee of the Russian Federation are charged with creation of a criminal community. Not only that, but their involvement in large bribes, money laundering, obstruction of justice, abuse of power, extortion, and racketeering has been reported. Due to the activities of these officers, dozens of high-profile cases including the ones against criminal lords had been ultimately ruined."

The choice of answers depends on context. Rhetoric structure allows differentiating between "official", "politically correct", template-based answers and "actual", "raw", "reports from the field", or "controversial" answers. (See FIG. **9** and FIG. **10**). Sometimes, the question itself can give a hint about which category of answers is expected. If a question is formulated as a factoid or definitional one, without a second meaning, then the first category of answers is suitable. Otherwise, if a question has the meaning "tell me what it really is", then the second category is appropriate. In general, after extracting a rhetoric structure from a question, selecting a suitable answer that would have a similar, matching, or complementary rhetoric structure is easier.

The official answer is based on elaboration and joints, which are neutral in terms of controversy a text might contain (See FIG. **9**). At the same time, the row answer includes the contrast relation. This relation is extracted between the phrase for what an agent is expected to do and what this agent was discovered to have done.

Classification of Request-Response Pairs

Application **102** can determine whether a given answer or response, such as an answer obtained from an answer database or a public database, is responsive to a given question, or request. More specifically, application **102** analyzes whether a request and response pair is correct or incorrect by determining one or both of (i) relevance or (ii) rhetoric agreement between the request and the response. Rhetoric agreement can be analyzed without taking into account relevance, which can be treated orthogonally.

Application **102** can determine similarity between question-answer pairs using different methods. For example, application **102** can determine level of similarity between an individual question and an individual answer. Alternatively, application **102** can determine a measure of similarity between a first pair including a question and an answer, and a second pair including a question and answer.

For example, application **102** uses classification model **120** trained to predict matching or non-matching answers. Application **102** can process two pairs at a time, for example <q1, a1> and <q2, a2>. Application **102** compares q1 with q2 and a1 with a1, producing a combined similarity score. Such a comparison allows a determination of whether an unknown question/answer pair contains a correct answer or not by assessing a distance from another question/answer pair with a known label. In particular, an unlabeled pair <q2, a2> can be processed so that rather than "guessing" correctness based on words or structures shared by q2 and a2, both q2 and a2 can be compared with their corresponding components q1 and a2 of the labeled pair <q2, a2> on the grounds of such words or structures. Because this approach targets a domain-independent classification of an answer, only the structural cohesiveness between a question and answer can be leveraged, not 'meanings' of answers.

In an aspect, application **102** uses training data **125** to train classification model **120**. In this manner, classification model **120** is trained to determine a similarity between pairs of questions and answers. This is a classification problem. Training data **125** can include a positive training set and a negative training set. Training data **125** includes matching request-response pairs in a positive dataset and arbitrary or lower relevance or appropriateness request-response pairs in a negative dataset. For the positive dataset, various domains with distinct acceptance criteria are selected that indicate whether an answer or response is suitable for the question.

Each training data set includes a set of training pairs. Each training set includes a question communicative discourse tree that represents a question and an answer communicative discourse tree that represents an answer and an expected level of complementarity between the question and answer. By using an iterative process, application **102** provides a training pair to classification model **120** and receives, from the model, a level of complementarity. Application **102** calculates a loss function by determining a difference between the determined level of complementarity and an expected level of complementarity for the particular training pair. Based on the loss function, application **102** adjusts internal parameters of the classification model to minimize the loss function.

Acceptance criteria can vary by application. For example, acceptance criteria may be low for community question answering, automated question answering, automated and manual customer support systems, social network communications and writing by individuals such as consumers about their experience with products, such as reviews and complaints. RR acceptance criteria may be high in scientific texts, professional journalism, health and legal documents in the form of FAQ, professional social networks such as "stackoverflow."

Communicative Discourse Trees (CDTs)

Application **102** can create, analyze, and compare communicative discourse trees. Communicative discourse trees are designed to combine rhetoric information with speech act structures. CDTs include with arcs labeled with expressions for communicative actions. By combining communicative actions, CDTs enable the modeling of RST relations and communicative actions. A CDT is a reduction of a parse thicket. See Galitsky, B, Ilvovsky, D. and Kuznetsov S O. Rhetoric Map of an Answer to Compound Queries Knowledge Trail Inc. ACL 2015, 681-686. ("Galitsky 2015"). A parse thicket is a combination of parse trees for sentences with discourse-level relationships between words and parts of the sentence in one graph. By incorporating labels that identify speech actions, learning of communicative discourse trees can occur over a richer features set than just rhetoric relations and syntax of elementary discourse units (EDUs).

In an example, a dispute between three parties concerning the causes of a downing of a commercial airliner, Malaysia Airlines Flight **16** is analyzed. An RST representation of the arguments being communicated is built. In the example, three conflicting agents, Dutch investigators, The Investigative Committee of the Russian Federation, and the self-proclaimed Donetsk People's Republic exchange their opinions on the matter. The example illustrates a controversial conflict where each party does all it can to blame its opponent. To sound more convincing, each party does not just produce its claim but formulates a response in a way to rebuff the claims of an opponent. To achieve this goal, each party attempts to match the style and discourse of the opponents' claims.

FIG. **11** illustrates a communicative discourse tree for a claim of a first agent in accordance with an aspect. FIG. **11** depicts communicative discourse tree **100**, which represents the following text: "Dutch accident investigators say that evidence points to pro-Russian rebels as being responsible for shooting down plane. The report indicates where the missile was fired from and identifies who was in control of the territory and pins the downing of MH16 on the pro-Russian rebels."

As can be seen from FIG. **11**, non-terminal nodes of CDTs are rhetoric relations, and terminal nodes are elementary discourse units (phrases, sentence fragments) which are the subjects of these relations. Certain arcs of CDTs are labeled with the expressions for communicative actions, including the actor agent and the subject of these actions (what is being communicated). For example, the nucleus node for elaboration relation (on the left) are labeled with say (Dutch, evidence), and the satellite with responsible(rebels, shooting down). These labels are not intended to express that the subjects of EDUs are evidence and shooting down but instead for matching this CDT with others for the purpose of finding similarity between them. In this case just linking these communicative actions by a rhetoric relation and not providing information of communicative discourse would be too limited way to represent a structure of what and how is being communicated. A requirement for an RR pair to have the same or coordinated rhetoric relation is too weak, so an agreement of CDT labels for arcs on top of matching nodes is required.

The straight edges of this graph are syntactic relations, and curvy arcs are discourse relations, such as anaphora, same entity, sub-entity, rhetoric relation and communicative actions. This graph includes much richer information than just a combination of parse trees for individual sentences. In addition to CDTs, parse thickets can be generalized at the level of words, relations, phrases and sentences. The speech actions are logic predicates expressing the agents involved in the respective speech acts and their subjects. The arguments of logical predicates are formed in accordance to respective semantic roles, as proposed by a framework such as VerbNet. See Karin Kipper, Anna Korhonen, Neville Ryant, Martha Palmer, A Large-scale Classification of English Verbs, Language Resources and Evaluation Journal, 42(1), pp. 21-40, Springer Netherland, 2008. and/or Karin Kipper Schuler, Anna Korhonen, Susan W. Brown, VerbNet overview, extensions, mappings and apps, Tutorial, NAACL-HLT 2009, Boulder, Colo.

FIG. **12** illustrates a communicative discourse tree for a claim of a second agent in accordance with an aspect. FIG. **12** depicts communicative discourse tree **1200**, which represents the following text: "The Investigative Committee of the Russian Federation believes that the plane was hit by a missile, which was not produced in Russia. The committee cites an investigation that established the type of the missile."

FIG. **13** illustrates a communicative discourse tree for a claim of a third agent in accordance with an aspect. FIG. **13** depicts communicative discourse tree **1300**, which represents the following text: "Rebels, the self-proclaimed Donetsk People's Republic, deny that they controlled the territory from which the missile was allegedly fired. It became possible only after three months after the tragedy to say if rebels controlled one or another town."

As can be seen from communicative discourse trees **1100-1300**, a response is not arbitrary. A response talks about the same entities as the original text. For example, communicative discourse trees **1200** and **1300** are related to communicative discourse tree **1100**. A response backs up a disagreement with estimates and sentiments about these entities, and about actions of these entities.

More specifically, replies of involved agent need to reflect the communicative discourse of the first, seed message. As a simple observation, because the first agent uses Attribution to communicate his claims, the other agents have to follow the suite and either provide their own attributions or attack the validity of attribution of the proponent, or both. To capture a broad variety of features for how communicative structure of the seed message needs to be retained in consecutive messages, pairs of respective CDTs can be learned.

To verify the agreement of a request-response, discourse relations or speech acts (communicative actions) alone are often insufficient. As can be seen from the example depicted in FIGS. **11-13**, the discourse structure of interactions between agents and the kind of interactions are useful. However, the domain of interaction (e.g., military conflicts or politics) or the subjects of these interactions, i.e., the entities, do not need to be analyzed.

Representing Rhetoric Relations and Communicative Actions

In order to compute similarity between abstract structures, two approaches are frequently used: (1) representing these structures in a numerical space, and express similarity as a number, which is a statistical learning approach, or (2) using a structural representation, without numerical space, such as trees and graphs, and expressing similarity as a maximal common sub-structure. Expressing similarity as a maximal common sub-structure is referred to as generalization.

Learning communicative actions helps express and understand arguments. Computational verb lexicons help support acquisition of entities for actions and provide a rule-based form to express their meanings. Verbs express the semantics of an event being described as well as the relational information among participants in that event, and project the syntactic structures that encode that information. Verbs, and in particular communicative action verbs, can be highly variable and can display a rich range of semantic behaviors. In response, verb classification helps a learning systems to deal with this complexity by organizing verbs into groups that share core semantic properties.

VerbNet is one such lexicon, which identifies semantic roles and syntactic patterns characteristic of the verbs in each class and makes explicit the connections between the syntactic patterns and the underlying semantic relations that can be inferred for all members of the class. See Karin Kipper, Anna Korhonen, Neville Ryant and Martha Palmer, *Language Resources and Evaluation*, Vol. 42, No. 1 (March

2008), at 21. Each syntactic frame, or verb signature, for a class has a corresponding semantic representation that details the semantic relations between event participants across the course of the event.

For example, the verb amuse is part of a cluster of similar verbs that have a similar structure of arguments (semantic roles) such as amaze, anger, arouse, disturb, and irritate. The roles of the arguments of these communicative actions are as follows: Experiencer (usually, an animate entity), Stimulus, and Result. Each verb can have classes of meanings differentiated by syntactic features for how this verb occurs in a sentence, or frames. For example, the frames for amuse are as follows, using the following key noun phrase (NP), noun (N), communicative action (V), verb phrase (VP), adverb (ADV):

NP V NP. Example: "The teacher amused the children." Syntax: Stimulus V Experiencer. Clause: amuse(Stimulus, E, Emotion, Experiencer), cause(Stimulus, E), emotional_state(result(E), Emotion, Experiencer).

NP V ADV-Middle. Example: "Small children amuse quickly." Syntax: Experiencer V ADV. Clause: amuse(Experiencer, Prop):—, property(Experiencer, Prop), adv(Prop).

NP V NP-PRO-ARB. Example "The teacher amused." Syntax Stimulus V. amuse(Stimulus, E, Emotion, Experiencer): cause(Stimulus, E), emotional_state(result(E), Emotion, Experiencer).

NP.cause V NP. Example "The teacher's dolls amused the children." syntax Stimulus <+genitive> ('s) V Experiencer. amuse(Stimulus, E, Emotion, Experiencer): cause(Stimulus, E),
emotional_state(during(E), Emotion, Experiencer).

NP V NP ADJ. Example "This performance bored me totally." syntax Stimulus V Experiencer Result. amuse(Stimulus, E, Emotion, Experiencer). cause(Stimulus, E), emotional_state(result(E), Emotion, Experiencer), Pred(result(E), Experiencer).

Communicative actions can be characterized into clusters, for example:

Verbs with Predicative Complements (Appoint, characterize, dub, declare, conjecture, masquerade, orphan, captain, consider, classify), Verbs of Perception (See, sight, peer).

Verbs of Psychological State (Amuse, admire, marvel, appeal), Verbs of Desire (Want, long).

Judgment Verbs (Judgment), Verbs of Assessment (Assess, estimate), Verbs of Searching (Hunt, search, stalk, investigate, rummage, ferret), Verbs of Social Interaction (Correspond, marry, meet, battle), Verbs of Communication (Transfer(message), inquire, interrogate, tell, manner(speaking), talk, chat, say, complain, advise, confess, lecture, overstate, promise). Avoid Verbs (Avoid), Measure Verbs, (Register, cost, fit, price, bill), Aspectual Verbs (Begin, complete, continue, stop, establish, sustain.

Aspects described herein provide advantages over statistical learning models. In contrast to statistical solutions, aspects use a classification system can provide a verb or a verb-like structure which is determined to cause the target feature (such as rhetoric agreement). For example, statistical machine learning models express similarity as a number, which can make interpretation difficult.

Representing Request-Response Pairs

Representing request-response pairs facilitates classification based operations based on a pair. In an example, request-response pairs can be represented as parse thickets. A parse thicket is a representation of parse trees for two or more sentences with discourse-level relationships between words and parts of the sentence in one graph. See Galitsky 2015. Topical similarity between question and answer can expressed as common sub-graphs of parse thickets. The higher the number of common graph nodes, the higher the similarity.

FIG. **14** illustrates parse thickets in accordance with an aspect. FIG. **14** depicts parse thicket **1400** including a parse tree for a request **1401**, and a parse tree for a corresponding response **1402**.

Parse tree **1401** represents the question "I just had a baby and it looks more like the husband I had my baby with. However it does not look like me at all and I am scared that he was cheating on me with another lady and I had her kid. This child is the best thing that has ever happened to me and I cannot imagine giving my baby to the real mom."

Response **1402** represents the response "Marital therapists advise on dealing with a child being born from an affair as follows. One option is for the husband to avoid contact but just have the basic legal and financial commitments. Another option is to have the wife fully involved and have the baby fully integrated into the family just like a child from a previous marriage."

FIG. **14** represents a greedy approach to representing linguistic information about a paragraph of text. The straight edges of this graph are syntactic relations, and curvy arcs are discourse relations, such as anaphora, same entity, sub-entity, rhetoric relation and communicative actions. The solid arcs are for same entity/sub-entity/anaphora relations, and the dotted arcs are for rhetoric relations and communicative actions. Oval labels in straight edges denote the syntactic relations. Lemmas are written in the boxes for the nodes, and lemma forms are written on the right side of the nodes.

Parse thicket **1400** includes much richer information than just a combination of parse trees for individual sentences. Navigation through this graph along the edges for syntactic relations as well as arcs for discourse relations allows to transform a given parse thicket into semantically equivalent forms for matching with other parse thickets, performing a text similarity assessment task. To form a complete formal representation of a paragraph, as many links as possible are expressed. Each of the discourse arcs produces a pair of thicket phrases that can be a potential match.

Topical similarity between the seed (request) and response is expressed as common sub-graphs of parse thickets. They are visualized as connected clouds. The higher the number of common graph nodes, the higher the similarity. For rhetoric agreement, common sub-graph does not have to be large as it is in the given text. However, rhetoric relations and communicative actions of the seed and response are correlated and a correspondence is required.

Generalization for Communicative Actions

A similarity between two communicative actions A_1 and A_2 is defined as a an abstract verb which possesses the features which are common between A_1 and A_2. Defining a similarity of two verbs as an abstract verb-like structure supports inductive learning tasks, such as a rhetoric agreement assessment. In an example, a similarity between the following two common verbs, agree and disagree, can be generalized as follows: agree^disagree=verb(Interlocutor, Proposed_action, Speaker),where Interlocution is the person who proposed the Proposed_action to the Speaker and to whom the Speaker communicates their response. Proposed_action is an action that the Speaker would perform if they were to accept or refuse the request or offer, and The Speaker is the person to whom a particular action has been proposed and who responds to the request or offer made.

In a further example, a similarity between verbs agree and explain is represented as follows: agree^explain=verb(Interlocutor, *, Speaker). The subjects of communicative actions are generalized in the context of communicative actions and are not be generalized with other "physical" actions. Hence, aspects generalize individual occurrences of communicative actions together with corresponding subjects.

Additionally, sequences of communicative actions representing dialogs can be compared against other such sequences of similar dialogs. In this manner, the meaning of an individual communicative action as well as the dynamic discourse structure of a dialogue is (in contrast to its static structure reflected via rhetoric relations) is represented. A generalization is a compound structural representation that happens at each level. Lemma of a communicative action is generalized with lemma, and its semantic role are generalized with respective semantic role.

Communicative actions are used by text authors to indicate a structure of a dialogue or a conflict. See Searle, J. R. 1969, Speech acts: an essay in the philosophy of language. London: Cambridge University Press. Subjects are generalized in the context of these actions and are not generalized with other "physical" actions. Hence, the individual occurrences of communicative actions together are generalized with their subjects, as well as their pairs, as discourse "steps."

Generalization of communicative actions can also be thought of from the standpoint of matching the verb frames, such as VerbNet. The communicative links reflect the discourse structure associated with participation (or mentioning) of more than a single agent in the text. The links form a sequence connecting the words for communicative actions (either verbs or multi-words implicitly indicating a communicative intent of a person).

Communicative actions include an actor, one or more agents being acted upon, and the phrase describing the features of this action. A communicative action can be described as a function of the form: verb (agent, subject, cause), where verb characterizes some type of interaction between involved agents (e.g., explain, confirm, remind, disagree, deny, etc.), subject refers to the information transmitted or object described, and cause refers to the motivation or explanation for the subject.

A scenario (labeled directed graph) is a sub-graph of a parse thicket G=(V,A), where V={action_1, action_2 . . . action_n} is a finite set of vertices corresponding to communicative actions, and A is a finite set of labeled arcs (ordered pairs of vertices), classified as follows:

Each arc action_i, $\text{action}_j \in A_{sequence}$ corresponds to a temporal precedence of two actions v_i, ag_i, s_i, c_i and v_j, ag_j, s_j, c_j that refer to the same subject, e.g., $s_j=s_i$ or different subjects. Each arc action_i, $\text{action} \in A_{cause}$ corresponds to an attack relationship between action_i and action_j indicating that the cause of action_i in conflict with the subject or cause of action_j.

Subgraphs of parse thickets associated with scenarios of interaction between agents have some distinguishing features. For example, (1) all vertices are ordered in time, so that there is one incoming arc and one outgoing arc for all vertices (except the initial and terminal vertices), (2) for $A_{sequence}$ arcs, at most one incoming and only one outgoing arc are admissible, and (3) for A_{cause} arcs, there can be many outgoing arcs from a given vertex, as well as many incoming arcs. The vertices involved may be associated with different agents or with the same agent (i.e., when he contradicts himself). To compute similarities between parse thickets and their communicative action, induced subgraphs, the subgraphs of the same configuration with similar labels of arcs and strict correspondence of vertices are analyzed.

The following similarities exist by analyzing the arcs of the communicative actions of a parse thicket: (1) one communicative action from with its subject from T1 against another communicative action with its subject from T2 (communicative action arc is not used), and (2) a pair of communicative actions with their subjects from T1 compared to another pair of communicative actions from T2 (communicative action arcs are used).

Generalizing two different communicative actions is based on their attributes. See (Galitsky et al 2013). As can be seen in the example discussed with respect to FIG. **14**, one communicative action from T1, cheating(husband, wife, another lady) can be compared with a second from T2, avoid(husband, contact(husband, another lady)). A generalization results in communicative_action(husband, *) which introduces a constraint on A in the form that if a given agent (=husband) is mentioned as a subject of CA in Q, he(she) should also be a subject of (possibly, another) CA in A. Two communicative actions can always be generalized, which is not the case for their subjects: if their generalization result is empty, the generalization result of communicative actions with these subjects is also empty.

Generalization of RST Relations

Some relations between discourse trees can be generalized, such as arcs that represent the same type of relation (presentation relation, such as antithesis, subject matter relation, such as condition, and multinuclear relation, such as list) can be generalized. A nucleus or a situation presented by a nucleus is indicated by "N." Satellite or situations presented by a satellite, are indicated by "S." "W" indicates a writer. "R" indicates a reader (hearer). Situations are propositions, completed actions or actions in progress, and communicative actions and states (including beliefs, desires, approve, explain, reconcile and others). Generalization of two RST relations with the above parameters is expressed as:

$$rst1(N1,S1,W1,R1)\hat{}rst2(N2,S2,W2,R2)=(rst1\ \hat{}rst2)(N1\ \hat{}N2,S1\hat{}S2,W1\ \hat{}W2,R1\ \hat{}R2).$$

The texts in N1, S1, W1, R1 are subject to generalization as phrases. For example, rst1^ rst2 can be generalized as follows: (1) if relation_type(rst1)!=relation_type(rst2) then a generalization is empty. (2) Otherwise, the signatures of rhetoric relations are generalized as sentences: sentence(N1, S1, W1, R1)^sentence(N2, S2, W2, R2). See Iruskieta, Mikel, Iria da Cunha and Maite Taboada. A qualitative comparison method for rhetorical structures: identifying different discourse structures in multilingual corpora. Lang Resources & Evaluation. June 2015, Volume 49, Issue 2.

For example, the meaning of rst-background^rst-enablement=(S increases the ability of R to comprehend an element in N)^(R comprehending S increases the ability of R to perform the action in N)=increase-VB the-DT ability-NN of-IN R-NN to-IN.

Because the relations rst-background^rst-enablement differ, the RST relation part is empty. The expressions that are the verbal definitions of respective RST relations are then generalized. For example, for each word or a placeholder for a word such as an agent, this word (with its POS) is retained if the word the same in each input phrase or remove the word if the word is different between these phrases. The resultant expression can be interpreted as a common meaning between the definitions of two different RST relations, obtained formally.

Two arcs between the question and the answer depicted in FIG. **14** show the generalization instance based on the RST relation "RST-contrast". For example, "I just had a baby" is a RST-contrast with "it does not look like me," and related to "husband to avoid contact" which is a RST-contrast with "have the basic legal and financial commitments." As can be seen, the answer need not have to be similar to the verb phrase of the question but the rhetoric structure of the question and answer are similar. Not all phrases in the answer must match phrases in question. For example, the phrases that do not match have certain rhetoric relations with the phrases in the answer which are relevant to phrases in question.

Building a Communicative Discourse Tree

FIG. **15** illustrates an exemplary process for building a communicative discourse tree in accordance with an aspect. Application **102** can implement process **1500**. As discussed, communicative discourse trees enable improved search engine results.

At block **1501**, process **1500** involves accessing a sentence including fragments. At least one fragment includes a verb and words and each word includes a role of the words within the fragment, and each fragment is an elementary discourse unit. For example, application **102** accesses a sentence such as "Rebels, the self-proclaimed Donetsk People's Republic, deny that they controlled the territory from which the missile was allegedly fired" as described with respect to FIG. **13**.

Continuing the example, application **102** determines that the sentence includes several fragments. For example, a first fragment is "rebels . . . deny." A second fragment is "that they controlled the territory." A third fragment is "from which the missile was allegedly fired." Each fragment includes a verb, for example, "deny" for the first fragment and "controlled" for the second fragment. Although, a fragment need not include a verb.

At block **1502**, process **1500** involves generating a discourse tree that represents rhetorical relationships between the sentence fragments. The discourse tree including nodes, each nonterminal node representing a rhetorical relationship between two of the sentence fragments and each terminal node of the nodes of the discourse tree is associated with one of the sentence fragments.

Continuing the example, application **102** generates a discourse tree as shown in FIG. **13**. For example, the third fragment, "from which the missile was allegedly fired" elaborates on "that they controlled the territory." The second and third fragments together relate to attribution of what happened, i.e., the attack cannot have been the rebels because they do not control the territory.

At block **1503**, process **1500** involves accessing multiple verb signatures. For example, application **102** accesses a list of verbs, e.g., from VerbNet. Each verb matches or is related to the verb of the fragment. For example, for the first fragment, the verb is "deny." Accordingly, application **102** accesses a list of verb signatures that relate to the verb deny.

As discussed, each verb signature includes the verb of the fragment and one or more of thematic roles. For example, a signature includes one or more of noun phrase (NP), noun (N), communicative action (V), verb phrase (VP), or adverb (ADV). The thematic roles describing the relationship between the verb and related words. For example "the teacher amused the children" has a different signature from "small children amuse quickly." For the first fragment, the verb "deny," application **102** accesses a list of frames, or verb signatures for verbs that match "deny." The list is "NP V NP to be NP," "NP V that S" and "NP V NP."

Each verb signature includes thematic roles. A thematic role refers to the role of the verb in the sentence fragment. Application **102** determines the thematic roles in each verb signature. Example thematic roles include actor, agent, asset, attribute, beneficiary, cause, location destination source, destination, source, location, experiencer, extent, instrument, material and product, material, product, patient, predicate, recipient, stimulus, theme, time, or topic.

At block **1504**, process **1500** involves determining, for each verb signature of the verb signatures, a number of thematic roles of the respective signature that match a role of a word in the fragment. For the first fragment, application **102** determines that the verb "deny" has only three roles, "agent", "verb" and "theme."

At block **1505**, process **1500** involves selecting a particular verb signature from the verb signatures based on the particular verb signature having a highest number of matches. For example, referring again to FIG. **13**, deny in the first fragment "the rebels deny . . . that they control the territory" is matched to verb signature deny "NP V NP", and "control" is matched to control (rebel, territory). Verb signatures are nested, resulting in a nested signature of "deny (rebel, control(rebel, territory))."

Building a Dialogue from an Arbitrary Text

Certain aspects leverage Rhetoric Structure Theory (RST) and other techniques to form questions that correspond with identified answers from within text. Examples of use cases include training classification models, generating training data in a second domain from available training data in a first domain, and generating and providing a virtual dialogue to a user device. Text of any style or genre can be used.

Rhetoric Structure Theory (RST) and other techniques can be used to identify answers and questions from text. As discussed, the flow of entities in text can be represented via a discourse tree that includes elementary discourse units (EDUs). Many rhetorical relations between elementary discourse units are binary anti-symmetric and specify which EDU has more important (nucleus) compared to less important (satellite). Therefore, a dialogue can be formed from text by identifying nucleus EDUs and satellite EDUs. In some cases, questions are identified from a satellite EDU and nucleus EDUs remain as background answers. These questions tend to be local in nature, e.g., to a region of text. In other cases, questions can be produced from nucleus EDUs, can be global in nature, applying to an entire text.

Discourse-based techniques can also be used to determine a suitable location within the text for a question to be inserted. For example, a marker can be inserted in the text at the point at which a nucleus EDU is finished and before a satellite EDU starts. A generated question can be inserted at this marker. The question "interrupts" the dialogue flow between a text author and a person asking a question. The text of the satellite EDU and possibly consecutive text then flows naturally as the answer to the question. The question is ideally related to the entity of the nucleus. But not all nuclei are relevant—text is interrupted to answer a question only if the question is suitable. For example, if conversion of a satellite elementary discourse unit into a question fails, if generalization of a question fails, or if verification of a question fails, then a question is not generated and inserted into the text.

The problem of building dialogue from text T is formulated as splitting it into a sequence of answers A=[A$_1$ . . . A$_n$] to form a dialogue:

$$[A_1,<Q_1,A_2>, \ldots ,<Q_{n-1},A_n>],$$

where A_i answers Q_{i-1} and possibly previous question, and $\cup A_i=T$ Q_{i-1} is derived from the whole or a part of A_i by linguistic means and generalization. FIG. **16** depicts an example of a suitable process.

FIG. **16** illustrates an exemplary process **1600** for identifying questions from a text in accordance with an aspect. Application **102** can implement process **1600**.

At block **1601**, process **1600** involves constructing a communicative discourse tree from text. Examples of text include electronic documents, books, and other online content. The text can include fragments that are elementary discourse units. In an example, application **102** accesses input text **110**.

As discussed, a communicative discourse tree includes a discourse tree with added communicative actions. An example of forming a discourse tree is discussed with respect to FIG. **15**. A discourse tree represents rhetorical relationships between the fragments of text. The discourse tree includes nodes, each nonterminal node representing a rhetorical relationship between two of the fragments, and each terminal node of the nodes of the discourse tree is associated with one of the fragments. For example purposes, process **1600** is discussed with respect to FIG. **17**.

FIG. **17** illustrates an exemplary communicative discourse tree in accordance with an aspect. FIG. **17** depicts communicative discourse tree **1700**, which includes rhetorical relations **1701-1704**; elementary discourse units **1710-1714**; questions **1720** and **1721**; and communicative actions **1730** and **1731**.

Communicative discourse tree **1700** represents the following text: "Theranos has struggled behind the scenes to turn the excitement over its technology into reality. At the end of 2014, the lab instrument developed as the linchpin of its strategy handled just a small fraction of the tests then sold to consumers, according to four former employees." Elementary discourse units **1710-1714** are leaf nodes of the communicative discourse tree. Each pair of elementary discourse units is connected via a rhetorical relation. Rhetorical relation **1701** is of type explanation, rhetorical relation **1702** of type elaboration, rhetorical relation **1702** of type attribution, and rhetorical relation **1702** of type background. As can be seen, for example, elementary discourse units **1710** and **1711** are connected via rhetorical relation "elaboration."

At block **1602**, process **1600** involves identifying, from the discourse tree, a terminal node including a nucleus elementary discourse unit. As discussed, a nucleus elementary discourse unit is dominant to a corresponding satellite elementary discourse unit. Each rhetorical relationship associates two elementary discourse units: a nucleus elementary discourse unit and a satellite elementary discourse unit. The nucleus elementary discourse unit is dominant to a corresponding satellite elementary discourse unit.

Either nuclei or satellite elementary discourse units can be used to form questions, depending on a type of question desired. To generate questions that relate to a large portion of the text corresponding to the communicative discourse tree, nuclei elementary discourse units are used.

Once the text is split into elementary discourse units, application **102** identifies which text fragments will serve as answers to questions by identifying one or more nucleus elementary discourse units. Using FIG. **17** as an example, application **102** identifies nucleus elementary discourse unit **1710**.

At block **1603**, process **1600** involves extracting, from the communicative discourse tree, a particular verb signature corresponding to the nucleus elementary discourse unit. With the nucleus elementary discourse unit identified, application **102** identifies the corresponding verb signature. As used herein, communicative actions, represented as verb signatures, can use the following syntax format:

verb(subject, ______)

struggle (T, excitement (technology))

Continuing the example, nucleus **1710** corresponds to communicative action **1730**, which is struggle(T, excitement (technology)). Communicative action **1730** is associated with the rhetorical relation of "elaboration." A verb signature, "excitement(technology)" is extracted from communicative action **1730**.

At block **1604**, process **1600** involves determining a subject from the verb signature and the nucleus elementary discourse unit. Continuing the example, application **102** identifies a subject from the verb signature associated with communicative action **1730**. As can be seen, the subject of the verb signature identified at block **1603**, "excitement (technology)" is "technology." "Excitement" is a predicate with "technology" as an argument. Therefore, "technology" is the subject. "Excitement" is associated with communicative action "excite". Different techniques can be used to determine the subject, such as parsing, table lookup, or database lookup.

At block **1605**, process **1600** involves forming a question fragment from the subject. Continuing the example, application **102** generates a question 'Was there an excitement about [Theranos] technology?" from the subject "technology." (Similarly, for the elementary discourse unit **1713**, communicative action develop(T, small-fraction), application **102** can generate a question 'Does Theranos only do a small fraction of tests?")

Different approaches can be used to form a question fragment. For example, forming a question can include constructing a parse tree from the nucleus elementary discourse unit. The parse tree includes nodes, e.g., for a noun phrase or a verb phrase. Application **102** selects, from the nodes, a node that represents either (i) a noun, (ii) a verb, or (iii) adjective. Application **102** can then replace the identified word with a question word. Examples of question words are what, where, whom, who, how, or whose. Other question words are possible. Table 3, below, lists some examples of how various parts of a parse tree can be replaced with question words.

TABLE 3

question word insertion
Using Rhetorical Relations to Guide Question Formation

Node(s) Deleted	Formed Question
Noun Phrase (NP) -	Noun and preposition of noun are replaced Example: "The president of US announced a . . . " Substitution: "The president of US" –> Who Q = Who announced . . . ?

TABLE 3-continued

question word insertion
Using Rhetorical Relations to Guide Question Formation

Node(s) Deleted	Formed Question
Verb Phrase (VP)	Verb and verb subject are replaced with what/when/where Example: The president of US announced a new series of projects to improve the security of state VP = announced a new series of projects . . . => Verb + Verb subject => announced what/when/where Substitution: Verb itself –> what Question: what did the president announce
Predicate + Subject	President announced => subject + predicate + VP (predicate) Wh part = VP-verb itself = for whom/to what audience/for what purpose Substitution: President announce for X, X–> What . . . ? Question: President announced for whom? President announced to which of his supporters President announced for what reason
Verb phrase + noun phrase	Formed question = noun phrase with Wh word Verb phrase = Announced on the summer evening Noun phrase = the summer evening Substitution: noun phrase => when/what/where Question: when did announce? What did announce, where did announce
Verb phrase, considering Verb Net roles	Phrase = "Announced about relationship with China" Verb Net (Announced): Who, To Whom, What, When, Where) Select role: When Question: When announced about relationship with China?
Communicative verb phrase with its subject	Phrase: said that the president announced Subject = the president announced Substitution "president" –> Who Question: who announced?

Possessive noun phrase	Phrase = President's announcement
	Substitution President's –> whose
	Question: Whose announcement?
	Phrase = President of China
	Substitution = President of China => president of what
	Q: President of what . . . announced?

FIG. **18** illustrates an exemplary parse tree in accordance with an aspect. FIG. **18** includes parse tree **1800**, which represents the sentence: "Joe packed his tools and materials neatly." Parse tree **1800**, generated from text, can be used form questions.

For example, application **102** builds a parse and then identifies nouns, verbs and adjectives in the parse tree. Application **102** can substitute nodes of the parse tree with a question word, thereby reducing the parse tree. For every selected node, application **102** selects an appropriate question word by using the following rules: substituting "who" or "what" for a noun, "what . . . do" for a verb, or "which way" or "how is" for an adjective. As can be seen in Table 4, in an example, the sentence can be transformed into the question "What did Joe pack neatly?" Application **102** replaces the noun "tools and materials" and the related word "his," leaving "Joe packed neatly." Application **102** then inserts the question word "what," resulting in "What did Joe pack neatly?"

TABLE 4

transforming a statement into a question
Forming a Question for "Joe packed his tools and materials neatly."

Node(s) Deleted	Resulting Question
Tools and materials (NNS)	What did Joe pack neatly
Joe (NNP), his (PRP$)	Who packed tools and materials neatly? Whose tools and materials were packed neatly?
Neatly (RB)	How did Joe pack?

In an aspect, a type of rhetorical relation that is connected to a particular nucleus discourse unit can be used as a hint to determine an appropriate question word. For example, the rhetorical relation "elaboration" forms a "what" question, a "background" relation forms a "what question," and a rhetorical relation "cause" yields a "why" question.

TABLE 5

rhetorical relation to question word mapping
Using Rhetorical Relations to Guide Selection of Question Words

Rhetorical Relation	Question Words
elaboration	"what"
background	"what"
cause	"why", "what caused" Example: nucleus "I fell on the pavement" and satellite "since it was wet after he rain." Question: "Why did I fall?"
attribution	"how" Example: "how do you know that? Example: "According to CNN, dogs eat cats as they do not like diversity in animals." Question: "how do . . . know that dogs eat cats?" "What makes you believe . . . ?" "From whom do you know . . . ?" "Who told that?"
Temporal sequence	After/before what . . . ? What happened after what?
Enablement	"Which way do you do . . . " "By which means?" "What did . . . do to . . . "
Condition	"On what condition?"
Concession	"But why?" "How come?"
Purpose	"For what?" "For which purpose?"

Continuing the example, returning to FIG. **17**, application **102** creates question **1720**. Question **1720** "Struggled for what?" uses the "what" question word and corresponds to elementary discourse unit **1710** and rhetorical relation **1702** ("elaboration"). Application **102** forms question **1721** "What's the role of instrument development?" which uses question word "what." Question **1721** corresponds to elementary discourse unit **1713** and rhetorical relation **1704** ("background"). In the case that multiple possible questions can be derived from a particular nucleus elementary discourse unit (e.g., one "what" question and another "why" question), one of the questions can be randomly chosen.

For example, application **102** examines nucleus elementary discourse unit **1710**, which states "Theranos has struggled." From elementary discourse unit **1710**, application **102** creates question "Struggled for what" by removing "Theranos," which is a noun, leaving "But what has struggled" and verb "has," leaving "but struggled."

In an aspect, nodes that are linked by co-references (e.g., pronouns) can be used and/or more complex selection rules can be applied. If a node is selected, then the node is removed from the parse tree.

Syntactic Verification

In an aspect, application **102** can verify the generated question fragment by aligning syntactic representations. An example of a syntactic representation is a syntactic (parse) tree. Syntactic representations (in conjunction with syntactic rules) allow for all the sentences in a given language to be generated). For example, application **102** can access a first syntactic representation of a question template. Application **102** creates a second syntactic representation of the question fragment. Application **102** then aligns the first syntactic representation with the second syntactic representation and substitutes the question fragment into the first syntactic representation. Alignment is necessary to make sure that the question being formed has a proper syntactic structure of a question. A questions dataset such as the Stanford. A Question Answering Dataset (SQuAD) can be used as reference. Alignment helps to ensure that the questions are not only semantically correct but are also syntactically appropriate.

Application **102** identifies a level of similarity between the question fragment and a template. Classification model **120** can be used. Application **102** determines that the level of similarity is greater than a threshold. When the level is greater than the threshold, then the question can be used (e.g., added to text corpus **105**).

Question Verification

In an aspect, to verify that the formed and modified question includes text that will generate a good answer, application **102** can apply an open-domain question-answer (Q/A) technique. For example, given the whole original text and a formed question, application **102** can verify that the answer is the elementary discourse unit from which this question was formed and that the answer does not correspond to another EDU. An incorrect text fragment can appear as an answer if the question was substantially distorted by generalization or web mining. An example of a suitable Q/A system is the Pavlov.ai deep learning system.

For example, application **102** can generate one or more keywords from the question fragment (e.g., "technology") and submit the keywords to a search engine. The search engine can generate results based on the keywords. Application **102** obtains a search result (e.g., a textual document or reference) from the search engine and compares the search result with the question fragment. The comparison can be accomplished by deriving a maximal common subtree of the question fragment and the search result.

In a further aspect, application **102** can obtain a semantically similar phrase by searching for candidate questions in a database. Application **102** can then merge the phrase with the candidate question. Dialogue generation, or generating a Natural Language question conditioned on an answer and the corresponding document, is related to general content (sentence) generation, but can result in questions that are less random in nature than questions generated by deep-learning techniques.

Referring back to FIG. **17**, application **102** creates an annotated corpus of text from discourse tree **1700**. The annotated corpus of text follows:

But Theranos has struggled . . . .

Struggled for what?

behind the scenes to turn the excitement over its technology into reality. At the end of 2014, the lab instrument developed as . . . .

What's the role of instrument development?

the linchpin of its strategy handled just a small fraction of the tests then sold to consumers, ...

Who said that?

according to four former employees."

Application **102** can use the annotated corpus of text for different applications. Alternatively, application **102** can output the questions and/or answers as discrete entities. Examples of suitable applications include training classification models, providing a virtual dialogue to a user, and generating training data in a second domain from available training data in a first domain.

In a further aspect, application **102** can refine a question to create a more natural-sounding question. For example, converting the satellite elementary discourse unit into a question might result in an unnatural question such as "the linchpin of its strategy handled just a small fraction of the tests then sold to whom?" Instead, application **102** phrases the question as "What does its strategy handle?"

More specifically, to improve the meaningfulness, interestingness and diversity of a formed and generalized question, application **102** can use web-mining techniques. Web-mining techniques involve forming a web search query from the original question (e.g., from the satellite elementary discourse unit) and attempts to find an expression from a web document as close to this question as possible and also from a reputable source or popular source. For example, application **102** queries an internet source for the question. Application **102** obtains search results from the internet source and iterates through the search results, scoring document titles, snippet sentences, and other expressions in found documents on the basis of their semantic similarity to the query.

Semantic similarity is assessed via the syntactic generalization score between the query and the search result. If a search result of sufficient similarity is found, then application **102** substitutes one or more entities of the question with new ones from the search result. For two words of the same part of speech (POS), their generalization is the same word with the POS. If the lemmas for the two words are different, but the POS is the same, then the POS remains in the result. If lemmas are the same, but the POS is different, lemma stays in the result. A lemma represents a word without the related part-of-speech information.

To illustrate this concept, consider an example of two natural language expressions. The meanings of the expressions are represented by logic formulas. The unification and anti-unification of these formulas are constructed. Some words (entities) are mapped to predicates, some are mapped into their arguments, and some other words do not explicitly occur in logic form representation but indicate the above instantiation of predicates with arguments.

Consider the following two sentences "camera with digital zoom" and "camera with zoom for beginners." To express the meanings, the following logic predicates are used:

camera(name_of_feature, type_of_users) and
zoom(type_of_zoom).

Note that this is a simplified example, and as such, may have a reduced number of arguments as compared to more typical examples. Continuing the example, the above expressions can be represented as:

camera(zoom(digital), AnyUser),
camera(zoom(AnyZoom), beginner)

According to the notation, variables (non-instantiated values, not specified in NL expressions) are capitalized. Given the above pair of formulas, unification computes their most general specialization camera(zoom(digital), beginner), and anti-unification computes their most specific generalization, camera(zoom(AnyZoom), AnyUser).

At the syntactic level, the expressions are subjected to a generalization ('^') of two noun phrases as: {NN-camera, PRP-with, [digital], NN-zoom [for beginners]}. The expressions in square brackets are eliminated because they occur in one expression but do not occur in the other. As a result, obtain{NN-camera, PRP-with, NN-zoom]}, which is a syntactic analog of semantic generalization, is obtained.

The purpose of an abstract generalization is to find commonality between portions of text at various semantic levels. Generalization operation occurs on the one or more levels. Examples of levels are paragraph level, sentence level, phrase level, and word level.

At each level (except word-level), individual words, the result of generalization of two expressions is a set of expressions. In such set, for each pair of expressions so that one is less general than other, the latter is eliminated. Generalization of two sets of expressions is a set of sets which are the results of pair-wise generalization of these expressions.

Only a single generalization exists for a pair of words: if words are the same in the same form, the result is a node with this word in this form. To involve word2vec models (Mikolov et al., 2015), compute generalization of two different words, the following rule is used. If subject1=subject2, then subject1^subject2=<subject1, POS (subject1), 1>. Otherwise, if they have the same part-of-speech, subject1^subject2=<*,POS(subject1), word2vecDistance(subject1^subject2)>. If part-of-speech is different, generalization is an empty tuple. It cannot be further generalized.

For a pair of phrases, generalization includes all maximum ordered sets of generalization nodes for words in phrases so that the order of words is retained. In the following example,

"To buy digital camera today, on Monday."

"Digital camera was a good buy today, first Monday of the month."

Generalization is {<JJ-digital, NN-camera>, <NN-today, ADV, Monday>}, where the generalization for noun phrases is followed by the generalization for an adverbial phrase. The verb buy is excluded from both generalizations because it occurs in a different order in the above phrases. Buy—digital—camera is not a generalization phrase because buy occurs in different sequence with the other generalization nodes.

Template Matching

In an aspect, application **102** can use template matching in conjunction with or instead of forming questions based on nucleus elementary discourse units. Template matching can be performed at block **1603**. Template matching involves identifying a level of similarity between the question fragment and a question template and determining that the level of similarity is greater than a threshold. An example of a question template is "What is a British rock band that formed in London?" Templates can help ensure that questions are a proper level of specificity. By contrast, 'What is a British rock band that formed in London in 1970 and received Grammy Hall of Fame Award in 2004?' would be too specific and should be reduced.

Templates can be generated offline. To achieve a proper level of generalization for questions, application **102** accesses an extended set of questions such as Stanford Q/A database (SQuAD), performs pair-wise syntactic generalization (Galitsky et al 2012) and retains most frequent question templates. SQUAD corpus (Rajpurkar et al., 2016) is a machine comprehension dataset consisting of over 100k crowd-sourced question-answer pairs on five hundred Wikipedia articles. For example, generalizing 'What is the purpose of life on Earth' and 'Tell me the purpose of complex numbers' application **102** obtains 'the-DT purpose-NN of-PRP*-NP' where we retain the part-of-speech tags. Application **102** collects the most frequent generalization results (question templates).

Templates obtained from extended questions can be further generalized to form additional questions. For example, application **102** can form, from additional text, a first parse tree from a first question and a second parse tree from a second question. An entity is identified from the first parse tree and the second parse tree. Application **102** generalizes the first parse tree and the second parse tree into a generalized fragment including the entity and adds the generalized fragment into the templates.

Application **102** can apply phrase-reduction rules at both individual phrase and sentence level. As a result, a question is obtained from an original nucleus EDU expression that is as close to a question template as possible. Hence for every nucleus EDU expression, application **102** iterates through the templates and find the most similar one. In terms of syntactic generalization, it is a template which delivers a maximal common sub-parse tree with this expression. For sentence "[I built a bridge]$_{nucleus}$ [with the purpose of fast access to the forest]$_{satellite}$", the satellite EDU is better covered by the template from our previous paragraph than, for example, by 'access-NN to-TO forest-NN' or 'access-NN to-TO NP' in terms of the number of common terms (parse tree nodes) of the generalization result.

Question Verification by System Testing

In an aspect, the generated question is provided to the system and the output answer is verified against an expected answer. For example, application **102** derives an answer that corresponds to the question from the nucleus elementary discourse unit. Application **102** can optionally train a classification model to generate the answer, or algorithmic techniques can be used. Training can including providing the questions, answers, or CDTs formed therefrom to the classification model.

Application **102** provides the question to the classification model and verifies that the answer is received from the classification model.

Annotated Text as Training Data

Certain aspects of the present disclosure use communicative discourse trees to generate robust training data sets, which can then be used to train one or more machine learning models. In turn, the machine learning models trained in this manner facilitate improved dialogue on the part of autonomous agents.

Certain aspects create a training set for a particular domain. In the case that training data for the particular domain is not available, then aspects can generate training data from arbitrary documents in that domain.

In other aspects, the training data can be used for transfer learning, that is, generating training data from text that is in a first domain (e.g., subject) and to train a model that is employed in a second domain. For example, training data based on a legal text could be used to train a model that is deployed in an autonomous agent that is used in banking. One advantage is that in some domains, obtaining training data may not be possible.

For example, if text is in a first domain, application **102** can generate training data in a second domain. Application **102** generating, from an utterance (or fragment of text), an additional communicative discourse tree. Application **102** applies a classification model to the additional communicative discourse tree. The classification model is trained with questions and answers. Application **102** receives an indication of whether the utterance is in rhetorical agreement with a reference text from the classification model. If the text is in rhetorical agreement, then the text is added to a training data set.

Annotated Text as Virtual Dialogue

Application **102** can create a virtual social dialogue from the questions and answers generated by process **1600**. A virtual social dialogue is a multi-turn dialogue between imaginary agents that is obtained as a result of content transformation. Application **102** can label each of the question fragments and answers with labels that identify a particular virtual actor. The application **102** can then provide the virtual conversation to the user device.

Global and Local Questions

Both global questions, (questions that can be answered by a text as a whole) and local questions (questions specific to a particular elementary discourse unit) can be formed. For example, referring back to the discourse tree **1700** in FIG. **17**, the following are the questions which can be answered by this text as a whole:

<What for/Why/When> did Theranos struggle? (Nucleus of the Elaboration)

<When/How/Why/What> was the lab instrument developed? (Nucleus for the Background).

FIG. **19** illustrates an exemplary communicative discourse tree that indicates identified questions from a text, in accordance with an aspect. FIG. **19** indicates elementary discourse units for generating global questions (i.e., for the whole text), which are marked by upper case "Q." These elementary discourse units are of type nucleus. Conversely, non-marked elementary discourse units indicate satellites that could be used to generate dialogue-level questions. Dialogue questions can be related to respective satellite EDUs (particular parts of text).

In FIG. **19**, we tag each EDU with respect to its role for forming questions: Nucleuses serve as 'global' questions Q such that the corresponding text (e.g., paragraph) forms an answer. Conversely, satellites can serve as 'local' questions q to form a dialogue, but not be asked expecting the whole text as an answer.

This split into global questions for the whole text and local questions for its individual parts via a dialogue is reliable as long as rhetorical parsing is correct. In the real life, some local questions would be good for the whole text, and some global questions would not be fully/adequately answered by this text; however, for the purpose of dialogue management, this split provides a reliable tool.

Within the disclosed framework, a dialogue can start with a global question, followed by local questions. Now, what happens in a real time when a user starts asking questions? We outline a dialogue management algorithm based on direct match of user utterances against the ones in the training set dialogues. We will compare this algorithm with a more popular, learning-based one.

An initial user question U_1 is matched with global questions $\{Q_g\}$, and a set of suitable dialogues S is formed, possibly with most relevant representative $D^* \in S$. Each such dialogue D_1 has its Q_g containing the same entities as U_1.

A first answer A1 from this representative is given. Once the user produces her second utterance U2, it is matched with the Q1 from the set of suitable dialogues. For A1 it can be D, a different dialogue from D* as long as its Qj or Aj, is most relevant to U2. Each consecutive Ui should be answered by D □ S with most relevant Qj or Aj.

FIG. **20** depicts a simplified diagram of a distributed system **2000** for implementing one of the aspects. In the illustrated aspect, distributed system **2000** includes one or more client computing devices **2002**, **2004**, **2006**, and **2008**, which are configured to execute and operate a client application such as a web browser, proprietary client (e.g., Oracle Forms), or the like over one or more network(s) **2010**. Server **2012** may be communicatively coupled with remote client computing devices **2002**, **2004**, **2006**, and **2008** via network **2010**.

In various aspects, server **2012** may be adapted to run one or more services or software applications provided by one or more of the components of the system. The services or software applications can include non-virtual and virtual environments. Virtual environments can include those used for virtual events, tradeshows, simulators, classrooms, shopping exchanges, and enterprises, whether two- or three-dimensional (3D) representations, page-based logical environments, or otherwise. In some aspects, these services may be offered as web-based or cloud services or under a

Software as a Service (SaaS) model to the users of client computing devices **2002**, **2004**, **2006**, and/or **2008**. Users operating client computing devices **2002**, **2004**, **2006**, and/or **2008** may in turn utilize one or more client applications to interact with server **2012** to utilize the services provided by these components.

In the configuration depicted in the figure, the software components **2018**, **2020** and **2022** of distributed system **2000** are shown as being implemented on server **2012**. In other aspects, one or more of the components of distributed system **2000** and/or the services provided by these components may also be implemented by one or more of the client computing devices **2002**, **2004**, **2006**, and/or **2008**. Users operating the client computing devices may then utilize one or more client applications to use the services provided by these components. These components may be implemented in hardware, firmware, software, or combinations thereof. It should be appreciated that various different system configurations are possible, which may be different from distributed system **2000**. The aspect shown in the figure is thus one example of a distributed system for implementing an aspect system and is not intended to be limiting.

Client computing devices **2002**, **2004**, **2006**, and/or **2008** may be portable handheld devices (e.g., an iPhone®, cellular telephone, an iPad®, computing tablet, a personal digital assistant (PDA)) or wearable devices (e.g., a Google Glass® head mounted display), running software such as Microsoft Windows Mobile®, and/or a variety of mobile operating systems such as iOS, Windows Phone, Android, BlackBerry 10, Palm OS, and the like, and being Internet, e-mail, short message service (SMS), Blackberry®, or other communication protocol enabled. The client computing devices can be general purpose personal computers including, by way of example, personal computers and/or laptop computers running various versions of Microsoft Windows®, Apple Macintosh®, and/or Linux operating systems. The client computing devices can be workstation computers running any of a variety of commercially-available UNIX® or UNIX-like operating systems, including without limitation the variety of GNU/Linux operating systems, such as for example, Google Chrome OS. Alternatively, or in addition, client computing devices **2002**, **2004**, **2006**, and **2008** may be any other electronic device, such as a thin-client computer, an Internet-enabled gaming system (e.g., a Microsoft Xbox gaming console with or without a Kinect® gesture input device), and/or a personal messaging device, capable of communicating over network(s) **2010**.

Although exemplary distributed system **2000** is shown with four client computing devices, any number of client computing devices may be supported. Other devices, such as devices with sensors, etc., may interact with server **2012**.

Network(s) **2010** in distributed system **2000** may be any type of network familiar to those skilled in the art that can support data communications using any of a variety of commercially-available protocols, including without limitation TCP/IP (transmission control protocol/Internet protocol), SNA (systems network architecture), IPX (Internet packet exchange), AppleTalk, and the like. Merely by way of example, network(s) **2010** can be a local area network (LAN), such as one based on Ethernet, Token-Ring and/or the like. Network(s) **2010** can be a wide-area network and the Internet. It can include a virtual network, including without limitation a virtual private network (VPN), an intranet, an extranet, a public switched telephone network (PSTN), an infra-red network, a wireless network (e.g., a network operating under any of the Institute of Electrical and Electronics (IEEE) 802.20 suite of protocols, Bluetooth®, and/or any other wireless protocol); and/or any combination of these and/or other networks.

Server **2012** may be composed of one or more general purpose computers, specialized server computers (including, by way of example, PC (personal computer) servers, UNIX® servers, mid-range servers, mainframe computers, rack-mounted servers, etc.), server farms, server clusters, or any other appropriate arrangement and/or combination. Server **2012** can include one or more virtual machines running virtual operating systems, or other computing architectures involving virtualization. One or more flexible pools of logical storage devices can be virtualized to maintain virtual storage devices for the server. Virtual networks can be controlled by server **2012** using software defined networking. In various aspects, server **2012** may be adapted to run one or more services or software applications described in the foregoing disclosure. For example, server **2012** may correspond to a server for performing processing described above according to an aspect of the present disclosure.

Server **2012** may run an operating system including any of those discussed above, as well as any commercially available server operating system. Server **2012** may also run any of a variety of additional server applications and/or mid-tier applications, including HTTP (hypertext transport protocol) servers, FTP (file transfer protocol) servers, CGI (common gateway interface) servers, JAVA® servers, database servers, and the like. Exemplary database servers include without limitation those commercially available from Oracle, Microsoft, Sybase, IBM (International Business Machines), and the like.

In some implementations, server **2012** may include one or more applications to analyze and consolidate data feeds and/or event updates received from users of client computing devices **2002**, **2004**, **2006**, and **2008**. As an example, data feeds and/or event updates may include, but are not limited to, Twitter® feeds, Facebook® updates or real-time updates received from one or more third party information sources and continuous data streams, which may include real-time events related to sensor data applications, financial tickers, network performance measuring tools (e.g., network monitoring and traffic management applications), clickstream analysis tools, automobile traffic monitoring, and the like. Server **2012** may also include one or more applications to display the data feeds and/or real-time events via one or more display devices of client computing devices **2002**, **2004**, **2006**, and **2008**.

Distributed system **2000** may also include one or more databases **2014** and **2016**. Databases **2014** and **2016** may reside in a variety of locations. By way of example, one or more of databases **2014** and **2016** may reside on a non-transitory storage medium local to (and/or resident in) server **2012**. Alternatively, databases **2014** and **2016** may be remote from server **2012** and in communication with server **2012** via a network-based or dedicated connection. In one set of aspects, databases **2014** and **2016** may reside in a storage-area network (SAN). Similarly, any necessary files for performing the functions attributed to server **2012** may be stored locally on server **2012** and/or remotely, as appropriate. In one set of aspects, databases **2014** and **2016** may include relational databases, such as databases provided by Oracle, that are adapted to store, update, and retrieve data in response to SQL-formatted commands.

FIG. **21** is a simplified block diagram of one or more components of a system environment **2100** by which services provided by one or more components of an aspect system may be offered as cloud services, in accordance with an aspect of the present disclosure. In the illustrated aspect, system environment **2100** includes one or more client computing devices **2104**, **2106**, and **2108** that may be used by users to interact with a cloud infrastructure system **2102** that provides cloud services. The client computing devices may be configured to operate a client application such as a web browser, a proprietary client application (e.g., Oracle Forms), or some other application, which may be used by a user of the client computing device to interact with cloud infrastructure system **2102** to use services provided by cloud infrastructure system **2102**.

It should be appreciated that cloud infrastructure system **2102** depicted in the figure may have other components than those depicted. Further, the aspect shown in the figure is only one example of a cloud infrastructure system that may incorporate an aspect of the disclosure. In some other aspects, cloud infrastructure system **2102** may have more or fewer components than shown in the figure, may combine two or more components, or may have a different configuration or arrangement of components.

Client computing devices **2104**, **2106**, and **2108** may be devices similar to those described above for **2002**, **2004**, **2006**, and **2008**.

Although exemplary system environment **2100** is shown with three client computing devices, any number of client computing devices may be supported. Other devices such as devices with sensors, etc. may interact with cloud infrastructure system **2102**.

Network(s) **2110** may facilitate communications and exchange of data between client computing devices **2104**, **2106**, and **2108** and cloud infrastructure system **2102**. Each network may be any type of network familiar to those skilled in the art that can support data communications using any of a variety of commercially-available protocols, including those described above for network(s) **2010**.

Cloud infrastructure system **2102** may comprise one or more computers and/or servers that may include those described above for server **2012**.

In certain aspects, services provided by the cloud infrastructure system may include a host of services that are made available to users of the cloud infrastructure system on demand, such as online data storage and backup solutions, Web-based e-mail services, hosted office suites and document collaboration services, database processing, managed technical support services, and the like. Services provided by the cloud infrastructure system can dynamically scale to meet the needs of its users. A specific instantiation of a service provided by cloud infrastructure system is referred to herein as a "service instance." In general, any service made available to a user via a communication network, such as the Internet, from a cloud service provider's system is referred to as a "cloud service." Typically, in a public cloud environment, servers and systems that make up the cloud service provider's system are different from the customer's own on-premises servers and systems. For example, a cloud service provider's system may host an application, and a user may, via a communication network such as the Internet, on demand, order and use the application.

In some examples, a service in a computer network cloud infrastructure may include protected computer network access to storage, a hosted database, a hosted web server, a software application, or other service provided by a cloud vendor to a user, or as otherwise known in the art. For example, a service can include password-protected access to remote storage on the cloud through the Internet. As another example, a service can include a web service-based hosted relational database and a script-language middleware engine for private use by a networked developer. As another example, a service can include access to an email software application hosted on a cloud vendor's web site.

In certain aspects, cloud infrastructure system **2102** may include a suite of applications, middleware, and database service offerings that are delivered to a customer in a self-service, subscription-based, elastically scalable, reliable, highly available, and secure manner. An example of such a cloud infrastructure system is the Oracle Public Cloud provided by the present assignee.

Large volumes of data, sometimes referred to as big data, can be hosted and/or manipulated by the infrastructure system on many levels and at different scales. Such data can include data sets that are so large and complex that it can be difficult to process using typical database management tools or traditional data processing applications. For example, terabytes of data may be difficult to store, retrieve, and process using personal computers or their rack-based counterparts. Such sizes of data can be difficult to work with using most current relational database management systems and desktop statistics and visualization packages. They can require massively parallel processing software running thousands of server computers, beyond the structure of commonly used software tools, to capture, curate, manage, and process the data within a tolerable elapsed time.

Extremely large data sets can be stored and manipulated by analysts and researchers to visualize large amounts of data, detect trends, and/or otherwise interact with the data. Tens, hundreds, or thousands of processors linked in parallel can act upon such data in order to present it or simulate external forces on the data or what it represents. These data sets can involve structured data, such as that organized in a database or otherwise according to a structured model, and/or unstructured data (e.g., emails, images, data blobs (binary large objects), web pages, complex event processing). By leveraging an ability of an aspect to relatively quickly focus more (or fewer) computing resources upon an objective, the cloud infrastructure system may be better available to carry out tasks on large data sets based on demand from a business, government agency, research organization, private individual, group of like-minded individuals or organizations, or other entity.

In various aspects, cloud infrastructure system **2102** may be adapted to automatically provision, manage and track a customer's subscription to services offered by cloud infrastructure system **2102**. Cloud infrastructure system **2102** may provide the cloud services via different deployment models. For example, services may be provided under a public cloud model in which cloud infrastructure system **2102** is owned by an organization selling cloud services (e.g., owned by Oracle) and the services are made available to the general public or different industry enterprises. As another example, services may be provided under a private cloud model in which cloud infrastructure system **2102** is operated solely for a single organization and may provide services for one or more entities within the organization. The cloud services may also be provided under a community cloud model in which cloud infrastructure system **2102** and the services provided by cloud infrastructure system **2102** are shared by several organizations in a related community. The cloud services may also be provided under a hybrid cloud model, which is a combination of two or more different models.

In some aspects, the services provided by cloud infrastructure system **2102** may include one or more services provided under Software as a Service (SaaS) category, Platform as a Service (PaaS) category, Infrastructure as a Service (IaaS) category, or other categories of services including hybrid services. A customer, via a subscription order, may order one or more services provided by cloud infrastructure system **2102**. Cloud infrastructure system **2102** then performs processing to provide the services in the customer's subscription order.

In some aspects, the services provided by cloud infrastructure system **2102** may include, without limitation, application services, platform services and infrastructure services. In some examples, application services may be provided by the cloud infrastructure system via a SaaS platform. The SaaS platform may be configured to provide cloud services that fall under the SaaS category. For example, the SaaS platform may provide capabilities to build and deliver a suite of on-demand applications on an integrated development and deployment platform. The SaaS platform may manage and control the underlying software and infrastructure for providing the SaaS services. By utilizing the services provided by the SaaS platform, customers can utilize applications executing on the cloud infrastructure system. Customers can acquire the application services without the need for customers to purchase separate licenses and support. Various different SaaS services may be provided. Examples include, without limitation, services that provide solutions for sales performance management, enterprise integration, and business flexibility for large organizations.

In some aspects, platform services may be provided by the cloud infrastructure system via a PaaS platform. The PaaS platform may be configured to provide cloud services that fall under the PaaS category. Examples of platform services may include without limitation services that enable organizations (such as Oracle) to consolidate existing applications on a shared, common architecture, as well as the ability to build new applications that leverage the shared services provided by the platform. The PaaS platform may manage and control the underlying software and infrastructure for providing the PaaS services. Customers can acquire the PaaS services provided by the cloud infrastructure system without the need for customers to purchase separate licenses and support. Examples of platform services include, without limitation, Oracle Java Cloud Service (JCS), Oracle Database Cloud Service (DBCS), and others.

By utilizing the services provided by the PaaS platform, customers can employ programming languages and tools supported by the cloud infrastructure system and also control the deployed services. In some aspects, platform services provided by the cloud infrastructure system may include database cloud services, middleware cloud services (e.g., Oracle Fusion Middleware services), and Java cloud services. In one aspect, database cloud services may support shared service deployment models that enable organizations to pool database resources and offer customers a Database as a Service in the form of a database cloud. Middleware cloud services may provide a platform for customers to develop and deploy various business applications, and Java cloud services may provide a platform for customers to deploy Java applications, in the cloud infrastructure system.

Various different infrastructure services may be provided by an IaaS platform in the cloud infrastructure system. The infrastructure services facilitate the management and control of the underlying computing resources, such as storage, networks, and other fundamental computing resources for customers utilizing services provided by the SaaS platform and the PaaS platform.

In certain aspects, cloud infrastructure system **2102** may also include infrastructure resources **2130** for providing the resources used to provide various services to customers of the cloud infrastructure system. In one aspect, infrastructure resources **2130** may include pre-integrated and optimized combinations of hardware, such as servers, storage, and networking resources to execute the services provided by the PaaS platform and the SaaS platform.

In some aspects, resources in cloud infrastructure system **2102** may be shared by multiple users and dynamically re-allocated per demand. Additionally, resources may be allocated to users in different time zones. For example, cloud infrastructure system **2102** may enable a first set of users in a first time zone to utilize resources of the cloud infrastructure system for a specified number of hours and then enable the re-allocation of the same resources to another set of users located in a different time zone, thereby maximizing the utilization of resources.

In certain aspects, a number of internal shared services **2132** may be provided that are shared by different components or modules of cloud infrastructure system **2102** and by the services provided by cloud infrastructure system **2102**. These internal shared services may include, without limitation, a security and identity service, an integration service, an enterprise repository service, an enterprise manager service, a virus scanning and white list service, a high availability, backup and recovery service, service for enabling cloud support, an email service, a notification service, a file transfer service, and the like.

In certain aspects, cloud infrastructure system **2102** may provide comprehensive management of cloud services (e.g., SaaS, PaaS, and IaaS services) in the cloud infrastructure system. In one aspect, cloud management functionality may include capabilities for provisioning, managing and tracking a customer's subscription received by cloud infrastructure system **2102**, and the like.

In one aspect, as depicted in the figure, cloud management functionality may be provided by one or more modules, such as an order management module **2120**, an order orchestration module **2122**, an order provisioning module **2124**, an order management and monitoring module **2126**, and an identity management module **2128**. These modules may include or be provided using one or more computers and/or servers, which may be general purpose computers, specialized server computers, server farms, server clusters, or any other appropriate arrangement and/or combination.

In exemplary operation **2134**, a customer using a client device, such as client computing device **2104**, **2106** or **2108**, may interact with cloud infrastructure system **2102** by requesting one or more services provided by cloud infrastructure system **2102** and placing an order for a subscription for one or more services offered by cloud infrastructure system **2102**. In certain aspects, the customer may access a cloud User Interface (UI) **2112**, cloud UI **2114** and/or cloud UI **2116** and place a subscription order via these UIs. The order information received by cloud infrastructure system **2102** in response to the customer placing an order may include information identifying the customer and one or more services offered by the cloud infrastructure system **2102** that the customer intends to subscribe to.

After an order has been placed by the customer, the order information is received via the cloud UIs, **2112**, **2114** and/or **2116**.

At operation **2136**, the order is stored in order database **2118**. Order database **2118** can be one of several databases operated by cloud infrastructure system **2102** and operated in conjunction with other system elements.

At operation **2138**, the order information is forwarded to an order management module **2120**. In some instances, order management module **2120** may be configured to perform billing and accounting functions related to the order, such as verifying the order, and upon verification, booking the order.

At operation **2140**, information regarding the order is communicated to an order orchestration module **2122**. Order orchestration module **2122** may utilize the order information to orchestrate the provisioning of services and resources for the order placed by the customer. In some instances, order orchestration module **2122** may orchestrate the provisioning of resources to support the subscribed services using the services of order provisioning module **2124**.

In certain aspects, order orchestration module **2122** enables the management of business processes associated with each order and applies business logic to determine whether an order should proceed to provisioning. At operation **2142**, upon receiving an order for a new subscription, order orchestration module **2122** sends a request to order provisioning module **2124** to allocate resources and configure those resources needed to fulfill the subscription order. Order provisioning module **2124** enables the allocation of resources for the services ordered by the customer. Order provisioning module **2124** provides a level of abstraction between the cloud services provided by cloud infrastructure system **2102** and the physical implementation layer that is used to provision the resources for providing the requested services. Order orchestration module **2122** may thus be isolated from implementation details, such as whether or not services and resources are actually provisioned on the fly or pre-provisioned and only allocated/assigned upon request.

At operation **2144**, once the services and resources are provisioned, a notification of the provided service may be sent to customers on client computing devices **2104**, **2106** and/or **2108** by order provisioning module **2124** of cloud infrastructure system **2102**.

At operation **2146**, the customer's subscription order may be managed and tracked by an order management and monitoring module **2126**. In some instances, order management and monitoring module **2126** may be configured to collect usage statistics for the services in the subscription order, such as the amount of storage used, the amount data transferred, the number of users, and the amount of system up time and system down time.

In certain aspects, cloud infrastructure system **2102** may include an identity management module **2128**. Identity management module **2128** may be configured to provide identity services, such as access management and authorization services in cloud infrastructure system **2102**. In some aspects, identity management module **2128** may control information about customers who wish to utilize the services provided by cloud infrastructure system **2102**. Such information can include information that authenticates the identities of such customers and information that describes which actions those customers are authorized to perform relative to various system resources (e.g., files, directories, applications, communication ports, memory segments, etc.) Identity management module **2128** may also include the management of descriptive information about each customer and about how and by whom that descriptive information can be accessed and modified.

FIG. **22** illustrates an exemplary computer system **2200**, in which various aspects of the present disclosure may be implemented. The computer system **2200** may be used to implement any of the computer systems described above. As shown in the figure, computer system **2200** includes a processing unit **2204** that communicates with a number of peripheral subsystems via a bus subsystem **2202**. These peripheral subsystems may include a processing acceleration unit **2206**, an I/O subsystem **2208**, a storage subsystem **2218** and a communications subsystem **2224**. Storage subsystem **2218** includes tangible computer-readable storage media **2222** and a system memory **2210**.

Bus subsystem **2202** provides a mechanism for letting the various components and subsystems of computer system **2200** communicate with each other as intended. Although bus subsystem **2202** is shown schematically as a single bus, alternative aspects of the bus subsystem may utilize multiple buses. Bus subsystem **2202** may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. For example, such architectures may include an Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus, which can be implemented as a Mezzanine bus manufactured to the IEEE P2286.1 standard.

Processing unit **2204**, which can be implemented as one or more integrated circuits (e.g., a conventional microprocessor or microcontroller), controls the operation of computer system **2200**. One or more processors may be included in processing unit **2204**. These processors may include single core or multicore processors. In certain aspects, processing unit **2204** may be implemented as one or more independent processing units **2232** and/or **2234** with single or multicore processors included in each processing unit. In other aspects, processing unit **2204** may also be implemented as a quad-core processing unit formed by integrating two dual-core processors into a single chip.

In various aspects, processing unit **2204** can execute a variety of programs in response to program code and can maintain multiple concurrently executing programs or processes. At any given time, some or all of the program code to be executed can be resident in processing units **2204** and/or in storage subsystem **2218**. Through suitable programming, processing units **2204** can provide various functionalities described above. Computer system **2200** may additionally include a processing acceleration unit **2206**, which can include a digital signal processor (DSP), a special-purpose processor, and/or the like.

I/O subsystem **2208** may include user interface input devices and user interface output devices. User interface input devices may include a keyboard, pointing devices such as a mouse or trackball, a touchpad or touch screen incorporated into a display, a scroll wheel, a click wheel, a dial, a button, a switch, a keypad, audio input devices with voice command recognition systems, microphones, and other types of input devices. User interface input devices may include, for example, motion sensing and/or gesture recognition devices such as the Microsoft Kinect® motion sensor that enables users to control and interact with an input device, such as the Microsoft Xbox® 360 game controller, through a natural user interface using gestures and spoken commands. User interface input devices may also include eye gesture recognition devices such as the Google Glass® blink detector that detects eye activity (e.g., 'blinking' while taking pictures and/or making a menu selection) from users and transforms the eye gestures as input into an input device (e.g., Google Glass®). Additionally, user interface input devices may include voice recognition sensing devices that enable users to interact with voice recognition systems (e.g., Siri® navigator), through voice commands.

User interface input devices may also include, without limitation, three dimensional (3D) mice, joysticks or pointing sticks, gamepads and graphic tablets, and audio/visual devices such as speakers, digital cameras, digital camcorders, portable media players, webcams, image scanners, fingerprint scanners, barcode reader 3D scanners, 3D printers, laser rangefinders, and eye gaze tracking devices. Additionally, user interface input devices may include, for example, medical imaging input devices such as computed tomography, magnetic resonance imaging, position emission tomography, medical ultrasonography devices. User interface input devices may also include, for example, audio input devices such as MIDI keyboards, digital musical instruments and the like.

User interface output devices may include a display subsystem, indicator lights, or non-visual displays such as audio output devices, etc. The display subsystem may be a cathode ray tube (CRT), a flat-panel device, such as that using a liquid crystal display (LCD) or plasma display, a projection device, a touch screen, and the like. In general, use of the term "output device" is intended to include all possible types of devices and mechanisms for outputting information from computer system **2200** to a user or other computer. For example, user interface output devices may include, without limitation, a variety of display devices that visually convey text, graphics and audio/video information such as monitors, printers, speakers, headphones, automotive navigation systems, plotters, voice output devices, and modems.

Computer system **2200** may comprise a storage subsystem **2218** that comprises software elements, shown as being currently located within a system memory **2210**. System memory **2210** may store program instructions that are loadable and executable on processing unit **2204**, as well as data generated during the execution of these programs.

Depending on the configuration and type of computer system **2200**, system memory **2210** may be volatile (such as random access memory (RAM)) and/or non-volatile (such as read-only memory (ROM), flash memory, etc.) The RAM typically contains data and/or program modules that are immediately accessible to and/or presently being operated and executed by processing unit **2204**. In some implementations, system memory **2210** may include multiple different types of memory, such as static random access memory (SRAM) or dynamic random access memory (DRAM). In some implementations, a basic input/output system (BIOS), containing the basic routines that help to transfer information between elements within computer system **2200**, such as during start-up, may typically be stored in the ROM. By way of example, and not limitation, system memory **2210** also illustrates application programs **2212**, which may include client applications, Web browsers, mid-tier applications, relational database management systems (RDBMS), etc., program data **2214**, and an operating system **2216**. By way of example, operating system **2216** may include various versions of Microsoft Windows®, Apple Macintosh®, and/or Linux operating systems, a variety of commercially-available UNIX® or UNIX-like operating systems (including without limitation the variety of GNU/Linux operating systems, the Google Chrome® OS, and the like) and/or mobile operating systems such as iOS, Windows® Phone, Android® OS, BlackBerry® 10 OS, and Palm® OS operating systems.

Storage subsystem **2218** may also provide a tangible computer-readable storage medium for storing the basic programming and data constructs that provide the functionality of some aspects. Software (programs, code modules, instructions) that when executed by a processor provide the functionality described above may be stored in storage subsystem **2218**. These software modules or instructions may be executed by processing unit **2204**. Storage subsystem **2218** may also provide a repository for storing data used in accordance with the present disclosure.

Storage subsystem **2218** may also include a computer-readable storage media reader **2220** that can further be connected to computer-readable storage media **2222**. Together and, optionally, in combination with system memory **2210**, computer-readable storage media **2222** may comprehensively represent remote, local, fixed, and/or removable storage devices plus storage media for temporarily and/or more permanently containing, storing, transmitting, and retrieving computer-readable information.

Computer-readable storage media **2222** containing code, or portions of code, can also include any appropriate media known or used in the art, including storage media and communication media, such as but not limited to, volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage and/or transmission of information. This can include tangible, non-transitory computer-readable storage media such as RAM, ROM, electronically erasable programmable ROM (EEPROM), flash memory or other memory technology, CD-ROM, digital versatile disk (DVD), or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or other tangible computer readable media. When specified, this can also include nontangible, transitory computer-readable media, such as data signals, data transmissions, or any other medium which can be used to transmit the desired information and which can be accessed by computer system **2200**.

By way of example, computer-readable storage media **2222** may include a hard disk drive that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive that reads from or writes to a removable, nonvolatile magnetic disk, and an optical disk drive that reads from or writes to a removable, nonvolatile optical disk such as a CD ROM, DVD, and Blu-Ray® disk, or other optical media. Computer-readable storage media **2222** may include, but is not limited to, Zip® drives, flash memory cards, universal serial bus (USB) flash drives, secure digital (SD) cards, DVD disks, digital video tape, and the like. Computer-readable storage media **2222** may also include, solid-state drives (SSD) based on non-volatile memory such as flash-memory based SSDs, enterprise flash drives, solid state ROM, and the like, SSDs based on volatile memory such as solid state RAM, dynamic RAM, static RAM, DRAM-based SSDs, magnetoresistive RAM (MRAM) SSDs, and hybrid SSDs that use a combination of DRAM and flash memory based SSDs. The disk drives and their associated computer-readable media may provide non-volatile storage of computer-readable instructions, data structures, program modules, and other data for computer system **2200**.

Communications subsystem **2224** provides an interface to other computer systems and networks. Communications subsystem **2224** serves as an interface for receiving data from and transmitting data to other systems from computer system **2200**. For example, communications subsystem **2224** may enable computer system **2200** to connect to one or more devices via the Internet. In some aspects, communications subsystem **2224** can include radio frequency (RF) transceiver components for accessing wireless voice and/or data networks (e.g., using cellular telephone technology, advanced data network technology, such as 3G, 4G or EDGE (enhanced data rates for global evolution), WiFi (IEEE 802.28 family standards, or other mobile communication technologies, or any combination thereof), global positioning system (GPS) receiver components, and/or other components. In some aspects, communications subsystem **2224** can provide wired network connectivity (e.g., Ethernet) in addition to or instead of a wireless interface.

In some aspects, communications subsystem **2224** may also receive input communication in the form of structured and/or unstructured data feeds **2226**, event streams **2228**, event updates **2230**, and the like on behalf of one or more users who may use computer system **2200**.

By way of example, communications subsystem **2224** may be configured to receive unstructured data feeds **2226** in real-time from users of social media networks and/or other communication services such as Twitter® feeds, Facebook® updates, web feeds such as Rich Site Summary (RSS) feeds, and/or real-time updates from one or more third party information sources.

Additionally, communications subsystem **2224** may also be configured to receive data in the form of continuous data streams, which may include event streams **2228** of real-time events and/or event updates **2230**, that may be continuous or unbounded in nature with no explicit end. Examples of applications that generate continuous data may include, for example, sensor data applications, financial tickers, network performance measuring tools (e.g. network monitoring and traffic management applications), clickstream analysis tools, automobile traffic monitoring, and the like.

Communications subsystem **2224** may also be configured to output the structured and/or unstructured data feeds **2226**, event streams **2228**, event updates **2230**, and the like to one or more databases that may be in communication with one or more streaming data source computers coupled to computer system **2200**.

Computer system **2200** can be one of various types, including a handheld portable device (e.g., an iPhone® cellular phone, an iPad® computing tablet, a PDA), a wearable device (e.g., a Google Glass® head mounted display), a PC, a workstation, a mainframe, a kiosk, a server rack, or any other data processing system.

Due to the ever-changing nature of computers and networks, the description of computer system **2200** depicted in the figure is intended only as a specific example. Many other configurations having more or fewer components than the system depicted in the figure are possible. For example, customized hardware might also be used and/or particular elements might be implemented in hardware, firmware, software (including applets), or a combination. Further, connection to other computing devices, such as network input/output devices, may be employed. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will appreciate other ways and/or methods to implement the various aspects.

In the foregoing specification, aspects of the disclosure are described with reference to specific aspects thereof, but those skilled in the art will recognize that the disclosure is not limited thereto. Various features and aspects of the above-described disclosure may be used individually or jointly. Further, aspects can be utilized in any number of environments and applications beyond those described herein without departing from the broader spirit and scope of the specification. The specification and drawings are, accordingly, to be regarded as illustrative rather than restrictive.

What is claimed is:

1. A computer-implemented method for constructing a question-answer association, the method comprising:

constructing, from text comprising fragments, a discourse tree that represents rhetorical relationships between the fragments, wherein the discourse tree comprises a plurality of nodes, each nonterminal node representing a rhetorical relationship between two of the fragments, and each terminal node of the nodes of the discourse tree is associated with one of the fragments, and wherein the fragments are elementary discourse units;

forming, from the discourse tree, a communicative discourse tree, wherein the forming comprises matching each fragment to a verb signature by:

accessing a plurality of verb signatures, wherein each verb signature comprises a verb of a fragment and a sequence of thematic roles, wherein each thematic role describes a respective relationship between the verb and related words;

determining, for each verb signature of the plurality of verb signatures, a plurality of thematic roles of the respective verb signature, wherein each of the thematic roles matches a role of a word in the fragment;

selecting a particular verb signature from the plurality of verb signatures based on the particular verb signature comprising a highest number of matches of roles of words to the verb; and

associating the particular verb signature with the fragment;

identifying, in the communicative discourse tree, a fragment that represents a nucleus elementary discourse unit, wherein the nucleus elementary discourse unit is dominant to a corresponding satellite elementary discourse unit;

extracting, from the communicative discourse tree, a particular verb signature corresponding to the nucleus elementary discourse unit;

determining a subject from the verb signature and the nucleus elementary discourse unit; and

forming, from the subject, a question fragment.

2. The computer-implemented method of claim **1**, wherein forming the question fragment comprises:

identifying, within the nucleus elementary discourse unit, a word that represents either (i) a noun, (ii) a verb, or (iii) adjective; and

replacing the word with a question word, wherein a question word is one of (i) what, (ii) where, (iii) whom, (iv) who, or (v) how.

3. The computer-implemented method of claim **2**, wherein identifying a word comprises:

constructing a parse tree from the nucleus elementary discourse unit, wherein the parse tree comprises nodes; and

selecting, from the nodes, a node that represents either (i) a noun, (ii) a verb, or (iii) adjective.

4. The computer-implemented method of claim **1**, wherein forming the question fragment comprises:

accessing a first syntactic representation of a question template;

aligning a second syntactic representation of the question fragment with the first syntactic representation; and

substituting the question fragment into the first syntactic representation.

5. The computer-implemented method of claim **1**, further comprising:

extracting a keyword from the question fragment;

submitting the keyword to a search engine;

obtaining a search result from the search engine;

comparing the search result with the question fragment by deriving a maximal common sub-tree of the question fragment and the search result;

responsive to determining that a number of words in the maximal common sub-tree is greater than a threshold:

aligning a second syntactic representation of the question fragment with a first syntactic representation of the search result; and

substituting the question fragment into the second syntactic representation.

6. The computer-implemented method of claim **1**, further comprising determining the plurality of templates by:

forming, from an additional text, a first parse tree from a first question and a second parse tree from a second question;

identifying, from the first parse tree, an entity;

identifying, from the second parse tree, the entity;

generalizing the first parse tree and the second parse tree into a generalized fragment comprising the entity; and

adding the generalized fragment into the plurality of templates.

7. The computer-implemented method of claim **1**, wherein the text is in a first domain, the method further comprising:

generating, from an utterance, an additional communicative discourse tree;

applying a classification model to the additional communicative discourse tree, wherein the classification model is trained with a plurality of questions and plurality of answers; and

receiving, from the classification model, an indication of whether the utterance is in rhetoric agreement with a reference text, wherein the reference text is in a second domain.

8. The computer-implemented method of claim **1**, further comprising:

deriving an answer from the nucleus elementary discourse unit;

training a classification model by inputting the question fragment and the answer to classification model;

providing the question fragment to the classification model; and

verifying that the answer is received from the classification model.

9. A non-transitory computer-readable medium storing computer-executable program instructions that when executed by a processing device, cause the processing device to perform operations comprising:

constructing, from text comprising fragments, a discourse tree that represents rhetorical relationships between the fragments, wherein the discourse tree comprises a plurality of nodes, each nonterminal node representing a rhetorical relationship between two of the fragments, and each terminal node of the nodes of the discourse tree is associated with one of the fragments, and wherein the fragments are elementary discourse units;

forming, from the discourse tree, a communicative discourse tree, wherein the forming comprises matching each fragment to a verb signature by:

accessing a plurality of verb signatures, wherein each verb signature comprises a verb of a fragment and a sequence of thematic roles, wherein each thematic role describes a respective relationship between the verb and related words;

determining, for each verb signature of the plurality of verb signatures, a plurality of thematic roles of the respective verb signature, wherein each of the thematic roles matches a role of a word in the fragment;

selecting a particular verb signature from the plurality of verb signatures based on the particular verb signature comprising a highest number of matches of roles of words to the verb; and

associating the particular verb signature with the fragment;

identifying, in the communicative discourse tree, a fragment that represents a nucleus elementary discourse unit, wherein the nucleus elementary discourse unit is dominant to a corresponding satellite elementary discourse unit;

extracting, from the communicative discourse tree, a particular verb signature corresponding to the nucleus elementary discourse unit;

determining a subject from the verb signature and the nucleus elementary discourse unit; and

forming, from the subject, a question fragment.

10. The non-transitory computer-readable medium of claim **9**, wherein forming the question fragment comprises:

identifying, within the nucleus elementary discourse unit, a word that represents either (i) a noun, (ii) a verb, or (iii) adjective; and

replacing the word with a question word, wherein a question word is one of (i) what, (ii) where, (iii) whom, (iv) who, or (v) how.

11. The non-transitory computer-readable medium of claim **9**, wherein identifying a word comprises:

constructing a parse tree from the nucleus elementary discourse unit, wherein the parse tree comprises nodes; and

selecting, from the nodes, a node that represents either (i) a noun, (ii) a verb, or (iii) adjective.

12. The non-transitory computer-readable medium of claim **9**, wherein forming the question fragment comprises:

accessing a first syntactic representation of a question template;

aligning a second syntactic representation of the question fragment with the first syntactic representation; and

substituting the question fragment into the first syntactic representation.

13. The non-transitory computer-readable medium of claim **9**, wherein the operations further comprise:

extracting a keyword from the question fragment;

submitting the keyword to a search engine;

obtaining a search result from the search engine;

comparing the search result with the question fragment by deriving a maximal common sub-tree of the question fragment and the search result;

responsive to determining that a number of words in the maximal common sub-tree is greater than a threshold:

aligning a second syntactic representation of the question fragment with a first syntactic representation of the search result; and

substituting the question fragment into the second syntactic representation.

14. The non-transitory computer-readable medium of claim **9**, wherein the operations further comprise determining the plurality of templates by:

forming, from an additional text, a first parse tree from a first question and a second parse tree from a second question;

identifying, from the first parse tree, an entity;

identifying, from the second parse tree, the entity;

generalizing the first parse tree and the second parse tree into a generalized fragment comprising the entity; and

adding the generalized fragment into the plurality of templates.

15. The non-transitory computer-readable medium of claim **9**, wherein the text is in a first domain, and wherein the operations further comprise:

generating, from an utterance, an additional communicative discourse tree;

applying a classification model to the additional communicative discourse tree, wherein the classification model is trained with a plurality of questions and plurality of answers; and

receiving, from the classification model, an indication of whether the utterance is in rhetoric agreement with a reference text, wherein the reference text is in a second domain.

16. A system comprising:

a non-transitory computer-readable medium storing computer-executable program instructions; and

a processing device communicatively coupled to the non-transitory computer-readable medium for executing the computer-executable program instructions, wherein executing the computer-executable program instructions configures the processing device to perform operations comprising:

constructing, from text comprising fragments, a discourse tree that represents rhetorical relationships between the fragments, wherein the discourse tree comprises a plurality of nodes, each nonterminal node representing a rhetorical relationship between two of the fragments, and each terminal node of the nodes of the discourse tree is associated with one of the fragments, and wherein the fragments are elementary discourse units;

forming, from the discourse tree, a communicative discourse tree, wherein the forming comprises matching each fragment to a verb signature by:

accessing a plurality of verb signatures, wherein each verb signature comprises a verb of a fragment and a sequence of thematic roles, wherein each thematic role describes a respective relationship between the verb and related words;

determining, for each verb signature of the plurality of verb signatures, a plurality of thematic roles of the respective verb signature, wherein each of the thematic roles matches a role of a word in the fragment;

selecting a particular verb signature from the plurality of verb signatures based on the particular verb signature comprising a highest number of matches of roles of words to the verb; and

associating the particular verb signature with the fragment;

identifying, in the communicative discourse tree, a fragment that represents a nucleus elementary discourse unit, wherein the nucleus elementary discourse unit is dominant to a corresponding satellite elementary discourse unit;

extracting, from the communicative discourse tree, a particular verb signature corresponding to the nucleus elementary discourse unit;

determining a subject from the verb signature and the nucleus elementary discourse unit; and

forming, from the subject, a question fragment.

17. The system of claim **16**, wherein forming the question fragment comprises:

identifying, within the nucleus elementary discourse unit, a word that represents either (i) a noun, (ii) a verb, or (iii) adjective; and

replacing the word with a question word, wherein a question word is one of (i) what, (ii) where, (iii) whom, (iv) who, or (v) how.

18. The system of claim **16**, wherein identifying a word comprises:

constructing a parse tree from the nucleus elementary discourse unit, wherein the parse tree comprises nodes; and

selecting, from the nodes, a node that represents either (i) a noun, (ii) a verb, or (iii) adjective.

19. The system of claim **16**, the operations further comprising:

extracting a keyword from the question fragment;

submitting the keyword to a search engine;

obtaining a search result from the search engine;

comparing the search result with the question fragment by deriving a maximal common sub-tree of the question fragment and the search result;

responsive to determining that a number of words in the maximal common sub-tree is greater than a threshold:

aligning a second syntactic representation of the question fragment with a first syntactic representation of the search result; and

substituting the question fragment into the second syntactic representation.

* * * * *