

```

1 # !pip install git+https://github.com/openai/swarm.git
2 # !pip install openai==1.55.3
3 # !pip install httpx==0.27.2
4 # !pip install --force-reinstall -v "openai==1.55.3"
5 # !pip install langchain==0.0.95
6 # !pip install langchain==0.1.20
7 # !pip install tiktoken==0.6.0
8 # !pip install langchain==0.1.20
9 # !pip install chromadb==0.5.0
10 # !pip install faiss-cpu
11 # !pip install PyPDF2
12 !pip install pypdf
13 # !pip install pymupdf

```

Collecting pypdf
 Downloading pypdf-5.1.0-py3-none-any.whl.metadata (7.2 kB)
 Requirement already satisfied: typing_extensions>=4.0 in /usr/local/lib/python3.10/dist-packages (from pypdf) (4.12.2)
 Downloading pypdf-5.1.0-py3-none-any.whl (297 kB)
 298.0/298.0 kB 4.4 MB/s eta 0:00:00
 Installing collected packages: pypdf
 Successfully installed pypdf-5.1.0

```

1 import os
2 # import httpx
3 from swarm import Swarm, Agent
4 from openai import OpenAI
5 from IPython.display import display
6 import ipywidgets as widgets
7 import time
8 import threading
9 import json
10 from langchain.embeddings import OpenAIEmbeddings
11 from langchain.vectorstores import FAISS
12 # from langchain.document_loaders import PyPDFLoader
13 from langchain.chains import RetrievalQA
14 from langchain.document_loaders import PyMuPDFLoader
15
16 from langchain.chat_models import ChatOpenAI
17 from langchain.document_loaders import CSVLoader
18 from langchain.embeddings import OpenAIEmbeddings
19 from langchain.prompts import ChatPromptTemplate
20 from langchain.vectorstores import Chroma
21 from langchain_core.output_parsers import StrOutputParser
22 from langchain_core.runnables import RunnableLambda, RunnablePassthrough
23 # from openai.embeddings_utils import get_embedding
24 import faiss
25 import PyPDF2
26
27 from langchain.document_loaders import PyPDFLoader
28 from langchain.vectorstores import FAISS
29 from langchain.embeddings import OpenAIEmbeddings
30 from langchain.chains import RetrievalQA
31 from langchain.llms import OpenAI
32 # from langchain_community.document_loaders import PyPDFLoader
33 # from langchain_community.vectorstores import FAISS
34 # from langchain_openai.embeddings import OpenAIEmbeddings
35 # from langchain.chains import RetrievalQA
36 # from langchain_openai.chat_models import ChatOpenAI
37 from langchain.chains.question_answering import load_qa_chain

```

```

1 os.environ['OPENAI_API_KEY'] = 'sk-proj-2xI-QA3u9MrI9Sqpt-lvl-kkIQbF0uSX5vZ7y7U7detvv0NsSwK264iz1u7Kv0dFyK6fyFeF9vT3B1bkFJ7
2 # openai.api_key = 'sk-proj-2xI-QA3u9MrI9Sqpt-lvl-kkIQbF0uSX5vZ7y7U7detvv0NsSwK264iz1u7Kv0dFyK6fyFeF9vT3B1bkFJ7b4Ni0UQnRVaT
3 api = OpenAI(api_key="sk-proj-3x6KqUMA4UwEhofqF3FPw7-kPNZKb5r0ry9tqPi27s3iSWltisFjT3ew6k3aVfBD2pFXi_c-I5T3B1bkFJ4xpGj0hvajm

```

✓ Swarm Setup

```
1 client = Swarm(api)
```

Triage Agent

History Taking Agent

```

1 # def handoff_to_medical_history_maker():
2 #     """Hand off the OLDCART history from the history taking agent to the medical history maker."""
3 #     return medical_history_maker_agent
4
5 def handoff_to_medical_history_maker(client, olcart_data):
6     """Sends OLDCART data to the Medical History Maker Agent."""
7     messages = [
8         {"role": "system", "content": (
9             "You are the Medical History Maker Agent. Take the structured OLDCART data "
10            "and produce a coherent medical history."
11        )},
12         {"role": "user", "content": json.dumps(olcart_data, indent=2)}
13     ]
14     response = client.run(agent=medical_history_maker_agent, messages=messages)
15     return response.messages[-1]["content"]
16
17
18 history_taking_agent = Agent(
19     name="History Taking Agent",
20     instructions="""
21     Relevance:
22     You are an expert medical assistant programmed to conduct detailed history-taking from patients presenting with medical
23     Your goal is to gather comprehensive and precise information to assist in clinical assessment.
24
25     Information:
26     Ask targeted, sequential questions to understand the patient's symptoms. Use the OLDCART framework to structure your ou
27
28     O: Onset
29     L: Location
30     D: Duration
31     C: Character
32     A: Aggravating factors
33     R: Relieving factors
34     T: Timing
35     Ensure all aspects are covered before concluding the history-taking.
36
37     Context:
38     You are interacting with a patient who presents with a symptom or complaint.
39     Based on their initial statement, refine your questions to explore each element of OLDCART systematically.
40
41     Constraints:
42     Be concise and avoid medical jargon that the patient may not understand.
43     Ensure all questions are respectful and maintain a conversational tone.
44     If the patient provides incomplete answers, prompt them with follow-up questions to clarify.
45
46     Examples:
47     Patient complaint: "I have a burning sensation when I urinate."
48
49     Output in OLDCART format:
50
51     Onset: When did the burning sensation begin? (e.g., "It started yesterday.")
52     Location: Where exactly do you feel the burning sensation? (e.g., "In the urinary tract.")
53     Duration: How long does the burning sensation last? (e.g., "Only while urinating.")
54     Character: How would you describe the sensation? (e.g., "It feels like burning pain.")
55     Aggravating factors: Does anything make it worse? (e.g., "Nothing specific.")
56     Relieving factors: Does anything make it better? (e.g., "Nothing helps.")
57     Timing: Does the sensation occur all the time or intermittently? (e.g., "Intermittently.")
58     Task: Begin by asking an open-ended question like, "What seems to be the problem today?" Then, guide the conversation t
59     Return a patient's history
60     """
61 ),
62 model = "gpt-4o-mini",
63 functions=[handoff_to_medical_history_maker],
64 )

```

```

1
2 def gather_history_with_OLDCART(agent, client):
3     # Extended OLDCART with severity and temporality
4     olcart_fields = {
5         "onset": "When did you first notice these symptoms starting?",
6         "location": "Where exactly do you feel these symptoms?",
7         "duration": "How long have these symptoms been occurring?",
8         "character": "How would you describe these symptoms (e.g., sharp, dull, burning)?",
9         "aggravating factors": "What seems to make these symptoms worse?",
10        "relieving factors": "Have you found anything that helps relieve these symptoms?",
11        "timing": "Do these symptoms occur at a specific time or are they constant?",
12        "severity": "On a scale of 0 to 10, how severe are these symptoms?",
13        "temporality": "Would you describe these symptoms as constant or intermittent?"
14    }
15
16    # Track completion status for each OLDCART field
17    olcart_fields_status = {field: False for field in olcart_fields.keys()}
18    # Store user responses in a dictionary for final formatting
19    olcart_responses = {field: None for field in olcart_fields.keys()}
20
21    def update_olcart_status(content):
22        # Simple logic: assign the current user input to the first unanswered field
23        for field in olcart_fields_status:
24            if not olcart_fields_status[field]:
25                olcart_fields_status[field] = True
26                olcart_responses[field] = content
27                break
28
29    # Initialize conversation structure
30    struct = []
31    initial_question = "How are you feeling today? Can you describe any symptoms or sickness you are experiencing?"
32    print(initial_question)
33    initial_response = input("Your response: ")
34    struct.append({"role": "user", "content": initial_response})
35
36    # Ask about each OLDCART field exactly once
37    for field, question in olcart_fields.items():
38        if not olcart_fields_status[field]:
39            # Prompt the agent for a question about this field
40            agent_prompt = (
41                f"The patient has described their symptoms, but we need clarification about '{field}'. "
42                "Ask one concise question to gather this detail without repeating other fields."
43            )
44            struct.append({"role": "system", "content": agent_prompt})
45            response = client.run(agent=agent, messages=struct)
46            agent_response = response.messages[-1]["content"]
47
48            # Remove the system message to maintain a clean conversation flow
49            struct.pop()
50
51            print(agent_response)
52            user_input = input("Your response: ")
53            struct.append({"role": "user", "content": user_input})
54
55            # Update OLDCART fields with the user's answer
56            update_olcart_status(user_input)
57
58    # Once done, print and also store the OLDCART history
59    print("OLDCART history completed!\n")
60    print("Patient History (OLDCART + Severity and Temporality):")
61    print(f"Onset: {olcart_responses['onset']}")
62    print(f"Location: {olcart_responses['location']}")
63    print(f"Duration: {olcart_responses['duration']}")
64    print(f"Character: {olcart_responses['character']}")
65    print(f"Aggravating factors: {olcart_responses['aggravating factors']}")
66    print(f"Relieving factors: {olcart_responses['relieving factors']}")
67    print(f"Severity: {olcart_responses['severity']}")
68    print(f"Temporality <constant | intermittent>: {olcart_responses['temporality']}")
69    print(f"Timing: {olcart_responses['timing']}\n")
70
71    # Add a final assistant message with structured OLDCART data
72    final_olcart_data = {
73        "onset": olcart_responses["onset"],
74        "location": olcart_responses["location"],
75        "duration": olcart_responses["duration"],
76        "character": olcart_responses["character"],

```

```

77     "aggravating factors": olcart_responses["aggravating factors"],
78     "relieving factors": olcart_responses["relieving factors"],
79     "severity": olcart_responses["severity"],
80     "temporality": olcart_responses["temporality"],
81     "timing": olcart_responses["timing"]
82 }
83
84 structured_message = {
85     "role": "assistant",
86     "content": json.dumps(final_olcart_data, indent=2)
87 }
88 struct.append(structured_message)
89
90 print("Conversation history:")
91 for message in struct:
92     print(f"{message['role']}.capitalize(): {message['content']}")
93
94 return struct

1 # Example usage
2 gathered_data = gather_history_with_OLDCART(agent=history_taking_agent, client=client)
3 print(gathered_data)

```

→ How are you feeling today? Can you describe any symptoms or sickness you are experiencing?
 Your response: I can't hear on my left ear
 When did you first notice that you couldn't hear in your left ear?
 Your response: 2 days ago
 Where do you feel the issue in your left ear – is it in the ear canal, eardrum area, or somewhere else?
 Your response: I think im allergic to my cat, ear canal
 How long have you been experiencing the hearing loss in your left ear?
 Your response: 2 days
 How would you describe the sensation in your left ear? For example, is it a feeling of fullness, pressure, or something else?
 Your response: fullness and pressure
 What, if anything, makes the fullness and pressure in your ear feel worse?
 Your response: i suppose when I jog
 What, if anything, helps relieve the fullness and pressure in your ear?
 Your response: sleeping on that side of my ear
 Does the fullness and pressure in your ear occur all the time, or does it come and go?
 Your response: all the time, i cant hear much
 On a scale of 1 to 10, how severe would you say your hearing loss is, with 1 being very mild and 10 being completely unable?
 Your response: 8
 Could you describe if the fullness and pressure in your ear varies throughout the day, or does it remain constant?
 Your response: remains constant throughout the day
 OLDCART history completed!

Patient History (OLDCART + Severity and Temporality):
 Onset: 2 days ago
 Location: I think im allergic to my cat, ear canal
 Duration: 2 days
 Character: fullness and pressure
 Aggravating factors: i suppose when I jog
 Relieving factors: sleeping on that side of my ear
 Severity: 8
 Temporality <constant | intermittent>: remains constant throughout the day
 Timing: all the time, i cant hear much

Conversation history:
 User: I can't hear on my left ear
 User: 2 days ago
 User: I think im allergic to my cat, ear canal
 User: 2 days
 User: fullness and pressure
 User: i suppose when I jog
 User: sleeping on that side of my ear
 User: all the time, i cant hear much
 User: 8
 User: remains constant throughout the day
 Assistant: {
 "onset": "2 days ago",
 "location": "I think im allergic to my cat, ear canal",
 "duration": "2 days",
 "character": "fullness and pressure",
 "aggravating factors": "i suppose when I jog",
 "relieving factors": "sleeping on that side of my ear",
 "severity": "8",
 "temporality": "remains constant throughout the day",
 "timing": "all the time, i cant hear much"
 }
 [{ 'role': 'user', 'content': "I can't hear on my left ear"}, { 'role': 'user', 'content': '2 days ago'}, { 'role': 'user', 'content': 'I think im allergic to my cat, ear canal'}, { 'role': 'user', 'content': '2 days'}, { 'role': 'user', 'content': 'fullness and pressure'}, { 'role': 'user', 'content': 'i suppose when I jog'}, { 'role': 'user', 'content': 'sleeping on that side of my ear'}, { 'role': 'user', 'content': 'all the time, i cant hear much'}, { 'role': 'user', 'content': '8'}, { 'role': 'user', 'content': 'remains constant throughout the day'}]

```

1 gathered_data
2 gathered_data = [{'role': 'user', 'content': "I can't hear on my left ear"},
3   {'role': 'user', 'content': '2 days ago'},
4   {'role': 'user', 'content': 'I think im allergic to my cat, ear canal '},
5   {'role': 'user', 'content': '2 days'},
6   {'role': 'user', 'content': 'fullness and pressure'},
7   {'role': 'user', 'content': 'i suppose when I jog'},
8   {'role': 'user', 'content': 'sleeping on that side of my ear'},
9   {'role': 'user', 'content': 'all the time, i cant hear much'},
10  {'role': 'user', 'content': '8'},
11  {'role': 'user', 'content': 'remains constant throughout the day'},
12  {'role': 'assistant',
13   'content': '{\n  "onset": "2 days ago",\n  "location": "I think im allergic to my cat, ear canal ",\n  "duration": "2 day

```

✓ Medical History Maker Agent

```

1 # def handoff_to_assessment_agent():
2 #     """Hand off the Patient Medical history to the Assessment Agent who will identify differentials diagnosis."""
3 #     return assessment_agent
4
5 def handoff_to_assessment_agent(client, medical_history):
6     """
7     Hand off the Patient Medical History to the Assessment Agent to identify differential diagnoses.
8     Args:
9         client: The client object for interacting with the agent.
10        medical_history: The structured medical history produced by the Medical History Maker Agent.
11    Returns:
12        The response from the Assessment Agent.
13    """
14    # Define the messages for the Assessment Agent
15    assessment_messages = [
16        {
17            "role": "system",
18            "content": (
19                "You are the Assessment Agent. Analyze the provided patient history and suggest "
20                "differential diagnoses. Provide a prioritized list with brief rationales for each."
21            )
22        },
23        {
24            "role": "user",
25            "content": medical_history # Pass the medical history as input
26        }
27    ]
28
29    # Run the Assessment Agent
30    response = client.run(agent=assessment_agent, messages=assessment_messages)
31    return response.messages[-1]["content"]
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

23 Incorporates a clearly identified chief complaint.
24 Incorporates all OLDCARTS elements, provided or elicited, in a coherent narrative.
25 Requires no further clarification from a clinical perspective, or if still ambiguous, has made a good faith effort to clarify.
26 Is well-organized, logically consistent, and easy for a clinician to read quickly.
27 Uses a factual, neutral tone without extraneous details or speculation.
28
29 Sample output could look something like the following:
30
31 Patient History:
32 Chief Complaint: The patient, a 35-year-old female, presents with a severe burning pain during urination.
33 History of Present Illness: The patient reports that the symptoms began less than one day ago. She describes the pain as a
34 Additional Information:
35 Past Medical History: The patient has no significant past medical history.
36 Medications: She is not currently taking any medications.
37 Allergies: The patient has no known allergies.
38 Social History: She is a non-smoker and consumes alcohol occasionally.
39 Family History: There is no family history of urinary tract infections.
40
41 """
42 ),
43 model = "gpt-4o-mini",
44 functions=[handoff_to_assessment_agent],
45 )

```

```

1 response = client.run(agent=medical_history_maker_agent, messages=gathered_data)
2 print(response.messages[0]["content"])

```

```

➦ Patient History:
Chief Complaint: The patient, a 30-year-old individual, presents with acute hearing loss in the left ear.

History of Present Illness: The patient reports that the symptoms began 2 days ago, describing a sensation of fullness and

Additional Information:
Past Medical History: Not provided.
Medications: Not provided.
Allergies: Possible allergy to cat dander.
Social History: Not provided.
Family History: Not provided.

```

TODO: fix hallucination if we need more info from patient

✓ Assessment Agent

```

1 # def handoff_to_treatment_agent():
2 #     """Hand off the Treatment Agent who will suggest treatment based on the diagnosis of the patient."""
3 #     return treatment_agent
4
5 def handoff_to_treatment_agent(client, assessment_output):
6     """
7     Hand off the Assessment Agent's output to the Treatment Agent for treatment recommendations.
8     Args:
9         client: The client object for interacting with the agent.
10        assessment_output: The output from the Assessment Agent containing differential diagnoses.
11    Returns:
12        The response from the Treatment Agent.
13    """
14    # Define the messages for the Treatment Agent
15    treatment_messages = [
16        {
17            "role": "system",
18            "content": (
19                "You are the Treatment Agent. Based on the differential diagnoses provided, "
20                "suggest appropriate next steps for patient management, including diagnostic tests, "
21                "initial treatments, or specialist referrals."
22            )
23        },
24        {
25            "role": "user",
26            "content": assessment_output # Pass the assessment results as input
27        }
28    ]
29

```

```

30 # Run the Treatment Agent
31 response = client.run(agent=treatment_agent, messages=treatment_messages)
32 return response.messages[-1]["content"]
33
34
1 assessment_agent = Agent(
2     name="Assessment Agent",
3     instructions="""
4 R (Role):
5 You are the Assessment Agent, a medical reasoning assistant trained to analyze patient histories and suggest potential diag
6
7 I (Input):
8 You will receive a detailed patient history, including symptoms, onset, location, duration, character, aggravating factors,
9 Your input may also include the patient's demographic information such as age, sex, and relevant lifestyle factors.
10
11 C (Context):
12 The Assessment Agent operates within a clinical reasoning framework. Your output is intended for healthcare professionals w
13
14 C (Constraints):
15
16 Clinical Accuracy: Ground your reasoning in standard medical knowledge and common clinical reasoning patterns. Avoid purely
17 Prioritization: Rank or highlight the most likely diagnoses first, explaining briefly why they are plausible given the pati
18 Clarity & Brevity: Present the differentials in a concise, organized manner. Include a short rationale for each primary opt
19 Non-Definitive: Do not provide a definitive diagnosis. Instead, present a range of reasonable considerations. It's acceptab
20 No Patient Management Instructions: Focus on the diagnostic reasoning aspect. Do not provide treatment plans or management
21 E (Evaluation):
22 Your output is successful if it:
23
24 Reflects a sound clinical reasoning process.
25 Lists likely differential diagnoses in an organized and prioritized fashion.
26 Uses medically appropriate terminology and is understandable by healthcare professionals.
27 Remains within the information provided and avoids unfounded speculation.
28 Is concise, coherent, and sufficiently explanatory to justify the presence of each suggested diagnosis.
29 """
30 ),
31 model = "gpt-4o-mini",
32 functions=[handoff_to_assessment_agent],
33 )

```

```


1 # usage
2
3 # Step 1: Gather OLDCART History from User via History Taking Agent
4 # gathered_data = gather_history_with_OLDCART(agent=history_taking_agent, client=client)
5
6 # The `gathered_data` structure now contains a series of messages between user and the agent.
7 # The last message should be the assistant message containing the OLDCART data in JSON.
8 final_olcart_data_str = None
9 for msg in reversed(gathered_data):
10     if msg["role"] == "assistant":
11         # Attempt to parse the content as JSON
12         try:
13             olcart_data = json.loads(msg["content"])
14             final_olcart_data_str = msg["content"] # Keep the raw JSON string to pass along
15             break
16         except json.JSONDecodeError:
17             # Not the structured OLDCART data, move on
18             continue
19
20 if not final_olcart_data_str:
21     raise ValueError("No structured OLDCART data found in the conversation.")
22
23 # Step 2: Assess the Patient's History Using the Assessment Agent
24 # Now we take the structured OLDCART data and pass it to the assessment agent
25 assessment_messages = [
26     {
27         "role": "system",
28         "content": (
29             "You are the Assessment Agent. Given the patient's OLDCARTS history data, "
30             "propose a list of possible differential diagnoses, in order of likelihood, "
31             "with brief rationales for each. Do not provide a definitive diagnosis. "
32             "If information is missing or ambiguous, note it, but do your best with the given data."
33         )
34     },
35     {
36         "role": "user",

```

```

37         "content": final_olcart_data_str # The JSON string from the history-taking step
38     }
39 ]
40
41 assessment_response = client.run(agent=assessment_agent, messages=assessment_messages)
42 assessment_output = assessment_response.messages[-1]["content"]
43 print("Assessment Agent Output:\n", assessment_output)
44
45 # The assessment_output should contain a list of differential diagnoses in some textual format.
46 # For better downstream usage, you might want the assessment agent to return a structured format (JSON),
47 # but here we assume it returns a readable text format.
48
49 # Step 3: Use the Treatment Agent to Suggest Next Steps Based on Differentials
50 # treatment_messages = [
51 #     {
52 #         "role": "system",
53 #         "content": (
54 #             "You are the Treatment Agent. Given the differential diagnoses provided, "
55 #             "suggest appropriate next steps, which could include further diagnostic testing, "
56 #             "initial management, or specialist referrals. Do not finalize a single diagnosis, "
57 #             "but guide the next reasonable clinical steps."
58 #         )
59 #     },
60 #     {
61 #         "role": "user",
62 #         "content": assessment_output # The text output from the assessment agent
63 #     }
64 # ]
65
66 # treatment_response = client.run(agent=treatment_agent, messages=treatment_messages)
67 # treatment_output = treatment_response.messages[-1]["content"]
68 # print("Treatment Agent Output:\n", treatment_output)
69
70 # At this point, you have:
71 # - A structured OLDCART history from the first agent
72 # - A set of possible differentials from the second agent
73 # - Suggested next steps from the treatment agent
74
75 # This completes the pipeline of connecting the three agents together.
76

```

 Assessment Agent Output:

Based on the provided OLDCARTS data, the following differential diagnoses are proposed for the patient's symptoms of fullness and pressure in the ears:

- Allergic Rhinitis / Eustachian Tube Dysfunction**
- **Rationale**: The initial suspicion of an allergy to the cat, combined with the sensation of fullness and pressure, is consistent with allergic rhinitis and Eustachian tube dysfunction.
- Otitis Media (Acute or Serous)**
- **Rationale**: The inability to hear much and the sensation of fullness could indicate fluid accumulation in the middle ear.
- External Otitis (Swimmer's Ear)**
- **Rationale**: Although more common in individuals with water exposure, inflammation or irritation in the ear canal could cause these symptoms.
- Sinusitis**
- **Rationale**: Since the patient has a history of allergy and may be experiencing referred pressure and fullness sensations, sinusitis is a possibility.
- Acute Allergic Reaction**
- **Rationale**: An acute allergic reaction potentially affecting the ear could also create a sensation of pressure and fullness.

Areas Requiring Further Information:

- History of Similar Symptoms**: Is this a recurrent issue or something new?
- Allergy Testing**: Confirmation of cat allergy or other potential allergens.
- Associated Symptoms**: Presence of nasal symptoms (sneezing, rhinorrhea) or systemic symptoms (rash, swelling).
- Visual Inspection**: An examination of the ear canal and tympanic membrane could clarify the presence of fluid or infection.

This proposed list of differential diagnoses offers a structured approach to understanding the patient's symptoms and high

```

1 def run_assessment_phase(gathered_data, assessment_agent, client):
2     # Extract the final OLDCART JSON data from the gathered_data
3     final_olcart_data_str = None
4     for msg in reversed(gathered_data):
5         if msg["role"] == "assistant":
6             try:
7                 # Attempt to parse the assistant message content as JSON
8                 json.loads(msg["content"])
9                 final_olcart_data_str = msg["content"]
10            break

```



```

11         except json.JSONDecodeError:
12             # Not structured OLDCART data, continue searching
13             continue
14
15     if not final_olcart_data_str:
16         raise ValueError("No structured OLDCART data found in the conversation.")
17
18     # Define the messages for the assessment agent
19     assessment_messages = [
20         {
21             "role": "system",
22             "content": (
23                 "You are the Doctor, Medical Assessment Agent. Given the patient's OLDCARTS history data, "
24                 "propose a list of possible differential diagnoses, in order of likelihood, "
25                 "with brief rationales for each. Do not provide a definitive diagnosis. "
26                 "If information is missing or ambiguous, note it, but do your best with the given data."
27             )
28         },
29         {
30             "role": "user",
31             "content": final_olcart_data_str
32         }
33     ]
34
35     # Run the assessment agent
36     assessment_response = client.run(agent=assessment_agent, messages=assessment_messages)
37     assessment_output = assessment_response.messages[-1]["content"]
38     print("Assessment Agent Output:\n", assessment_output)
39
40     return assessment_output

```

1 # Example usage:

```

2 # gathered_data = gather_history_with_OLDCART(agent=history_taking_agent, client=client)
3 assessment_output = run_assessment_phase(gathered_data, assessment_agent, client)
4 # assessment_output now contains the differential diagnoses from the Assessment Agent
5

```

➔ Assessment Agent Output:

Based on the provided OLDCARTS history, here is a list of potential differential diagnoses, ranked in order of likelihood

1. **Allergic Otitis Media**: The patient's symptoms of fullness and pressure in the ear canal, along with a suspected allergy, suggest this condition.
2. **Eustachian Tube Dysfunction (ETD)**: This condition can cause a sensation of fullness and pressure in the ear and is often associated with allergies.
3. **Earwax Impaction**: The symptoms of hearing loss, fullness, and pressure could indicate a blockage caused by cerumen.
4. **Acute Allergic Rhinitis**: This could be contributing to sinus or ear pressure due to inflammation from allergens. Allergies are a common cause of these symptoms.
5. **Chronic Otitis Media**: Although the duration of symptoms is only two days, the pressure and hearing loss could be signs of a chronic infection.
6. **Temporomandibular Joint Disorder (TMJ)**: If the ear fullness and pressure are coupled with jaw discomfort or changes in bite, TMJ could be a contributing factor.

Missing information that could refine the differential includes:

- Presence of other symptoms (nasal congestion, itchiness, etc.)
- History of similar issues or previous ear infections
- Any other known allergies other than the cat
- Medication usage or past medical history, particularly related to allergies or ear issues.

Further evaluation, including otoscopic examination and allergy testing, would be helpful to clarify the diagnosis.

✧ Treatment Agent

```

1 # def handoff_to_medication_agent():
2 #     """Hand off the Medication List Agent, a RAG model who will suggest what meds to take based on the assessment of the
3 #     return medication_agent
4
5 def handoff_to_medication_agent(client, medication_agent, treatment_output, medication_list_agent_query):
6     """
7     Handoff treatment recommendations to the Medication List Agent for medication suggestions.
8
9     Args:
10         client: The client object for interacting with the agent.
11         medication_agent: The Medication Agent definition.

```

```

12     treatment_output: Output from the Treatment Agent (treatment recommendations).
13     medication_list_agent_query: The RAG-based query function to retrieve medication information.
14
15 Returns:
16     str: The response from the Medication Agent containing the medication list.
17 """
18 print("Handoff: Sending treatment recommendations to Medication List Agent...\n")
19
20 # Step 1: Use the RAG query function to retrieve medication information
21 rag_query_result = medication_list_agent_query(
22     f"Based on the following treatment recommendations, identify relevant medications, their "
23     f"dosages, common side effects, and contraindications:\n{treatment_output}"
24 )
25
26 # Step 2: Define the input messages for the Medication Agent
27 medication_messages = [
28     {
29         "role": "system",
30         "content": (
31             "You are the Medication List Agent. Use the retrieved data below to generate a "
32             "comprehensive medication list, including dosages, side effects, and contraindications."
33         )
34     },
35     {
36         "role": "user",
37         "content": rag_query_result # Pass the RAG query result to the agent
38     }
39 ]
40
41 # Step 3: Run the Medication Agent
42 response = client.run(agent=medication_agent, messages=medication_messages)
43 return response.messages[-1]["content"]
44
45
46 1 treatment_agent = Agent(
47     2     name="Treatment Agent",
48     3     instructions="""
49     4 R (Role):
50     5 You are the Treatment Agent, acting in the role of a medical doctor. Your task is to recommend appropriate next steps in pa
51     6
52     7 I (Input):
53     8 You will be given the assessment results from the Assessment Agent, which detail a list of possible differential diagnoses
54     9
55     10 C (Context):
56     11 The context is a clinical environment where a patient has presented with certain symptoms and an Assessment Agent has provi
57     12
58     13 C (Constraints):
59     14
60     15 Evidence-Based and Safe: Provide standard-of-care, reasonable steps without overreaching.
61     16 No Definitive Diagnosis: Do not finalize a single, definitive diagnosis; maintain the differential nature.
62     17 Focus on Next Steps: Suggest further diagnostic tests, physical exams, initial treatments, lifestyle modifications, or refe
63     18 Acknowledge Uncertainty: If certain information is missing, note what would be helpful to obtain next.
64     19 Concise and Professional: Keep the guidance straightforward, actionable, and medically appropriate.
65     20 E (Evaluation):
66     21 Your output is successful if it:
67     22
68     23 Reflects a clear, clinically logical set of next steps.
69     24 Aligns with standard medical reasoning and patient care pathways.
70     25 Avoids speculation beyond the provided differentials.
71     26 Offers a concise, clinically relevant plan that another medical professional would find helpful.
72     27
73     28 """
74     29 ),
75     30 model = "gpt-4o-mini",
76     31 functions=[handoff_to_medication_agent],
77     32 )
78
79
80 1 # Step 3: Use the Treatment Agent with the Assessment Output
81 2 treatment_messages = [
82     3     {
83     4         "role": "system",
84     5         "content": (
85     6             "You are the Treatment Agent, acting as a medical doctor. "
86     7             "Given the patient's differential diagnoses from the Assessment Agent, "
87     8             "propose initial next steps or management strategies. "
88     9             "These could include further testing, conservative treatments, referrals, "

```

```

10         "or suggestions for additional information needed."
11     )
12 },
13 {
14     "role": "user",
15     "content": assessment_output
16 }
17 ]
18
19 treatment_response = client.run(agent=treatment_agent, messages=treatment_messages)
20 treatment_output = treatment_response.messages[-1]["content"]
21 print("Treatment Agent Output:\n", treatment_output)


```

Treatment Agent Output:

Based on the provided differential diagnoses and the information given in the OLDCARTS history for the patient, the following

1. **Physical Examination**:
 - Perform a thorough otoscopic examination to assess for signs of earwax impaction, signs of otitis media, or other structural abnormalities.
 2. **Allergy Assessment**:
 - Consider referring the patient for allergy testing to confirm the suspected cat allergy, which could be contributing to the symptoms.
 3. **Eustachian Tube Function**:
 - Assess Eustachian tube function through techniques like the Valsalva maneuver, which can help in evaluating if the Eustachian tubes are patent and functioning correctly.
 4. **Management Options**:
 - For **Allergic Otitis Media or Eustachian Tube Dysfunction**:
 - Initiate symptomatic treatment with antihistamines (e.g., cetirizine or loratadine) to manage allergy symptoms.
 - Recommend nasal corticosteroids (e.g., fluticasone) if allergic rhinitis is confirmed or suspected.
 - For **Earwax Impaction**:
 - If earwax impaction is confirmed, consider safe removal methods (e.g., irrigation or manual extraction) or recommending over-the-counter ceruminolytics.
 5. **Symptomatic Relief**:
 - Advise the patient on symptomatic relief strategies, including decongestants or saline nasal sprays if there are signs of nasal congestion.
 6. **Observation and Follow-Up**:
 - Since symptoms are only two days old and may resolve with initial management, observe the patient after initiating treatment for improvement in symptoms. Schedule a follow-up assessment in 1-2 weeks to evaluate further if there are no improvements.
 7. **Education**:
 - Educate the patient about the potential exacerbation of symptoms due to allergens and encourage avoidance of known triggers (like exposure to the cat) as much as possible.
 8. **Document Additional Information**:
 - Collect further history regarding any concurrent symptoms, past medical history, and medication use during the follow-up visit; this can help refine the diagnosis and tailor future management.
- Overall, the goal is to clarify the diagnosis while managing the symptoms effectively, considering the high suspicion of an allergic component in this case.

1 treatment_response

 Response(messages=[{'content': "Based on the provided differential diagnoses and the information given in the OLDCARTS history for the patient, the following next steps for management are recommended:\n\n1. **Physical Examination**: Perform a thorough otoscopic examination to assess for signs of earwax impaction, signs of otitis media, or other structural abnormalities. This will help clarify whether the ear's appearance correlates with the symptoms described.\n\n2. **Allergy Assessment**: Consider referring the patient for allergy testing to confirm the suspected cat allergy, which could be contributing to allergic otitis media or Eustachian tube dysfunction. Identifying specific allergens may guide management strategies.\n\n3. **Eustachian Tube Function**: Assess Eustachian tube function through techniques like the Valsalva maneuver, which can help in evaluating if the Eustachian tubes are patent and functioning correctly.\n\n4. **Management Options**: For **Allergic Otitis Media or Eustachian Tube Dysfunction**: Initiate symptomatic treatment with antihistamines (e.g., cetirizine or loratadine) to manage allergy symptoms. Recommend nasal corticosteroids (e.g., fluticasone) if allergic rhinitis is confirmed or suspected. For **Earwax Impaction**: If earwax impaction is confirmed, consider safe removal methods (e.g., irrigation or manual extraction) or recommending over-the-counter ceruminolytics.\n\n5. **Symptomatic Relief**: Advise the patient on symptomatic relief strategies, including decongestants or saline nasal sprays if there are signs of nasal congestion.\n\n6. **Observation and Follow-Up**: Since symptoms are only two days old and may resolve with initial management, observe the patient after initiating treatment for improvement in symptoms. Schedule a follow-up assessment in 1-2 weeks to evaluate further if there are no improvements.\n\n7. **Education**: Educate the patient about the potential exacerbation of symptoms due to allergens and encourage avoidance of known triggers (like exposure to the cat) as much as possible.\n\n8. **Document Additional Information**: Collect further history regarding any concurrent symptoms, past medical history, and medication use during the follow-up visit; this can help refine the diagnosis and tailor future management.\n\nOverall, the goal is to clarify the diagnosis while managing the symptoms effectively, considering the high suspicion of allergic components in this case.", 'refusal': None, 'role': 'assistant', 'audio': None, 'function_call': None, 'tool_calls': None, 'sender': 'Treatment Agent'}], agent=Agent(name='Treatment Agent', model='gpt-4o-mini', instructions='\nR (Role):\nYou are the Treatment Agent, acting in the role of a medical doctor. Your task is to recommend appropriate next steps in patient management based on the differential diagnoses provided.\n\nI (Input):\nYou will be given the assessment results from the Assessment Agent, which detail a list of possible differential diagnoses and their rationales. You must use this information to guide initial management and clinical decision-making.\n\nC (Context):\nThe context is a clinical environment where a patient has presented with certain symptoms and an Assessment Agent has provided a set of likely differential diagnoses. Your job is not to definitively diagnose the patient but to offer reasonable, evidence-based next steps that a physician might take to clarify, evaluate, or manage the patient's condition. These steps might include ordering specific tests, recommending conservative management measures, or suggesting referrals to

specialists if appropriate.\n\nC (Constraints):\n\nEvidence-Based and Safe: Provide standard-of-care, reasonable steps without overreaching.\n\nNo Definitive Diagnosis: Do not finalize a single, definitive diagnosis; maintain the differential nature.\n\nFocus on Next Steps: Suggest further diagnostic tests, physical exams, initial treatments, lifestyle modifications, or referrals as would be appropriate in a clinical setting.\n\nAcknowledge Uncertainty: If certain information is missing, note what would be helpful to obtain next.\n\nConcise and Professional: Keep the guidance straightforward, actionable, and medically appropriate.\n\nE (Evaluation):\n\nYour output is successful if it:\n\nReflects a clear, clinically logical set of next steps.\n\nAligns with standard medical reasoning and patient care pathways.\n\nAvoids speculation beyond the provided differentials.\n\nOffers a concise, clinically relevant plan that another medical professional would find helpful.\n\n', functions=[<function handoff_to_medication_agent at 0x7cbc7c5d7490>], tool_choice=None, parallel_tool_calls=True), context_variables={})

✓ Medication List Agent + RAG Model

```
1 def setup_pdf_qa_system(pdf_path: str, openai_api_key: str):
2     """
3     Sets up a QA system by processing a PDF document, creating embeddings,
4     storing them in a vector store, and initializing a retrieval-based QA chain.
5
6     Args:
7         pdf_path (str): Path to the PDF document.
8         openai_api_key (str): OpenAI API key for embedding and LLM services.
9
10    Returns:
11        function: A function that takes a query string and returns the answer.
12    """
13    # Initialize OpenAI embeddings
14    embeddings = OpenAIEmbeddings(openai_api_key=openai_api_key)
15
16    # Load and split the PDF document
17    loader = PyPDFLoader(pdf_path)
18    print("Loaded medication list...")
19    documents = loader.load()
20
21    # Create embeddings and store them in a FAISS vector store
22    vectorstore = FAISS.from_documents(documents, embeddings)
23    print("Vector store created...")
24
25    # Initialize the OpenAI language model
26    llm = OpenAI(openai_api_key=openai_api_key)
27    print("OpenAI model initialized...")
28
29    # Load a QA chain
30    combine_documents_chain = load_qa_chain(llm, chain_type="stuff")
31    print("Create QA chain...")
32
33    # Set up the retrieval-based QA chain
34    retrieval_chain = RetrievalQA(
35        retriever=vectorstore.as_retriever(),
36        combine_documents_chain=combine_documents_chain
37    )
38
39    def answer_query(query: str) -> str:
40        """Uses the QA system to answer a query."""
41        return retrieval_chain.run(query)
42
43    return answer_query
```

```
1 # openai_api_key = os.environ['OPENAI_API_KEY']
2 openai_api_key = "sk-proj-3x6KqUMa4UwEhofqF3FPw7-kPNZKb5r0ry9tqPi27s3iSWltisFjT3ew6k3aVfBD2pFXi_c-I5T3BlbkFJ4xpGj0hvajmGsGZ"
3 pdf_path = "/content/medication_list_edited_unstructured.pdf"
4 # Set up the QA system
5 qa_system = setup_pdf_qa_system(pdf_path, openai_api_key)
6 # Example query
7 query = "What are the recommended medications for hypertension?"
8
9 # Get the answer
10 answer = qa_system(query)
11 print(answer)
```

Loaded medication list...
Vector store created...
OpenAI model initialized...
Create QA chain...

The recommended medications for hypertension are anti-hypertensives, including diuretics, adrenergic inhibitors, ACE inhil

```

1 def handoff_to_medication_agent(medication_agent_query_function, treatment_output):
2     """
3     Uses the Medication List Agent to query the RAG model for medication details based on treatment output.
4     Args:
5         medication_agent_query_function: Function that queries the RAG model for medications.
6         treatment_output: Output from the Treatment Agent containing next steps or conditions.
7     Returns:
8         The medication list generated by the Medication List Agent.
9     """
10    print("Orchestrator: Passing treatment output to Medication List Agent for medication suggestions...\n")
11    query = (
12        "Based on the following treatment recommendations, identify relevant medications, their "
13        "dosages, common side effects, and contraindications:\n"
14        f"{treatment_output}"
15    )
16    return medication_agent_query_function(query)
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
26
```

```

13 C (Constraints):
14 - **Accuracy:** Base all information on the retrieved data from the knowledge base. Do not speculate or provide information
15 - **Structure:** Provide a structured list of medications with the following details:
16   1. **Medication Name**
17   2. **Dosage**
18   3. **Route of Administration**
19   4. **Common Side Effects**
20   5. **Contraindications**
21 - **Relevance:** Focus solely on medications relevant to the input treatment recommendations or conditions.
22 - **Professional Tone:** Use clear, concise, and medically appropriate language.
23 - **No Speculation:** If certain information is missing, acknowledge the gap rather than guessing or assuming.
24
25 E (Evaluation):
26 Your output will be considered successful if it:
27 1. Lists medications that are clinically relevant to the input treatment plan.
28 2. Includes accurate and comprehensive details about each medication (e.g., dosage, side effects, contraindications).
29 3. Avoids errors or omissions that could compromise patient safety.
30 4. Is structured in a clear, organized format suitable for medical professionals.
31
32 ---
33
34 ### **Example Input:**
35 "The patient requires treatment for a urinary tract infection (UTI). Recommend appropriate antibiotics and symptomatic relief."
36
37 ---
38
39 ### **Example Output:**
40 1. **Nitrofurantoin**
41   - **Dosage:** 100 mg, oral, twice daily for 5 days.
42   - **Route of Administration:** Oral.
43   - **Common Side Effects:** Nausea, headache, dizziness.
44   - **Contraindications:** Severe renal impairment (eGFR < 30 mL/min).
45
46 2. **Trimethoprim-Sulfamethoxazole**
47   - **Dosage:** 160/800 mg, oral, twice daily for 3 days.
48   - **Route of Administration:** Oral.
49   - **Common Side Effects:** Rash, gastrointestinal upset, hyperkalemia.
50   - **Contraindications:** Sulfa allergy, third-trimester pregnancy.
51
52 3. **Phenazopyridine**
53   - **Dosage:** 200 mg, oral, three times daily for 2 days.
54   - **Route of Administration:** Oral.
55   - **Common Side Effects:** Orange discoloration of urine, headache.
56   - **Contraindications:** Severe renal insufficiency.
57
58 ---
59
60 Use this structure and adhere strictly to the RICCE framework when generating medication lists.""
61
62     ),
63     model = "gpt-4o-mini",
64     functions=[handoff_to_prescription_agent],
65 )

```

✓ Basic Prescription Writer Agent

```

1 def handoff_to_medication_agent():
2     """Hand off the Medication List Agent, a RAG model who will suggest what meds to take based on the assessment of the patient"""
3     return medication_agent

1 prescription_agent = Agent(
2     name="Prescription Agent",
3     instructions="""
4 R (Role):
5 You are the Prescription Agent, an advanced medical assistant specializing in crafting personalized medication plans based on patient history and current treatment recommendations.
6
7 I (Input):
8 You will receive:
9 1. Treatment recommendations from the Treatment Agent.
10 2. A medication list generated by the Medication List Agent, including details such as dosage, side effects, and contraindications.
11 3. Optional contextual information about the patient, such as allergies, comorbidities, or other relevant clinical factors.
12
13 C (Context):

```

```

14 You are operating in a clinical environment where the final prescription plan must align with evidence-based medical guidelines
15
16 C (Constraints):
17 1. Accuracy: Ensure that all medication recommendations are consistent with the retrieved data and safe for the patient
18 2. Structure: Provide a clear prescription plan for each medication, including:
19     - Medication Name
20     - Dosage
21     - Route of Administration
22     - Frequency
23     - Duration
24     - Any special instructions or warnings (e.g., take with food, avoid alcohol).
25 3. Relevance: Only include medications that are directly relevant to the provided treatment recommendations.
26 4. Safety: Highlight any contraindications, interactions, or safety warnings explicitly.
27 5. Professional Tone: Use clear, medically appropriate language.
28
29 E (Evaluation):
30 Your output will be considered successful if it:
31 1. Includes an accurate and comprehensive prescription plan.
32 2. Aligns with the retrieved data and treatment recommendations.
33 3. Is structured in a professional and easy-to-read format.
34 4. Addresses any safety concerns (e.g., contraindications, interactions).
35
36 ---
37
38 ## Example Input:
39 1. Treatment Recommendations:
40     - Start empiric antibiotic therapy for urinary tract infection (UTI).
41     - Provide symptomatic relief for pain during urination.
42 2. Medication List from the Medication List Agent:
43     1. Nitrofurantoin
44         - Dosage: 100 mg, twice daily for 5 days.
45         - Side Effects: Nausea, dizziness.
46         - Contraindications: Renal impairment (eGFR < 30 mL/min).
47     2. Phenazopyridine
48         - Dosage: 200 mg, three times daily for 2 days.
49         - Side Effects: Orange discoloration of urine, headache.
50         - Contraindications: Severe renal disease.
51
52 ---
53
54 ## Example Output:
55 Prescription Plan:
56 1. Nitrofurantoin
57     - Dosage: 100 mg.
58     - Route of Administration: Oral.
59     - Frequency: Twice daily.
60     - Duration: 5 days.
61     - Special Instructions: Take with food to reduce nausea. Avoid in patients with renal impairment (eGFR < 30 mL/min).
62
63 2. Phenazopyridine
64     - Dosage: 200 mg.
65     - Route of Administration: Oral.
66     - Frequency: Three times daily.
67     - Duration: 2 days.
68     - Special Instructions: Take with plenty of water. Warn the patient about orange discoloration of urine.
69
70 ---
71
72 Adhere strictly to this format and structure for each case you process.
73     """),
74     model="gpt-4o-mini",
75     functions=[] # No functions for further handoffs; this is the final agent in the workflow.
76 )
77

```

✧ Orchestrator Triage Agent

```

1 def handoff_to_prescription_agent(client, medication_agent, treatment_output, medication_list_agent_query):
2     """
3     Handoff treatment recommendations to the Medication Agent for medication suggestions.
4
5     Args:
6         client: The client object for interacting with the Medication Agent.

```

```

7 medication_agent: The Medication Agent definition.
8 treatment_output: The treatment recommendations from the Treatment Agent.
9 medication_list_agent_query: The RAG-based query function to retrieve medication information.
10
11 Returns:
12     The response from the Medication Agent containing the medication list.
13 """
14 print("Orchestrator: Passing treatment recommendations to Medication Agent...\n")
15
16 # Use the RAG-based query function to retrieve medication information
17 rag_query_result = medication_list_agent_query(
18     f"Based on the following treatment recommendations, identify relevant medications, their "
19     f"dosages, common side effects, and contraindications:\n{treatment_output}"
20 )
21
22 # Define the input messages for the Medication Agent
23 medication_messages = [
24     {
25         "role": "system",
26         "content": (
27             "You are the Medication List Agent. Use the retrieved data below to generate a "
28             "comprehensive medication list, including dosages, side effects, and contraindications."
29         )
30     },
31     {
32         "role": "user",
33         "content": rag_query_result # Pass the RAG result as input to the agent
34     }
35 ]
36
37 # Run the Medication Agent
38 response = client.run(agent=medication_agent, messages=medication_messages)
39 return response.messages[-1]["content"]
40
1
2 # 2. Define Orchestrator Agent (Meta Agent)
3 orchestrator_agent = Agent(
4     name="Orchestrator Agent",
5     instructions="""
6 R (Role):
7 You are the Orchestrator Agent, overseeing and coordinating the flow of information
8 and tasks between the History Taking Agent, Assessment Agent, Treatment Agent, and Medication List Agent.
9
10 I (Input):
11 You will receive a trigger to start the patient workflow. Your job is to call the agents
12 in sequence to collect and process data: History → Assessment → Treatment → Medications.
13
14 C (Context):
15 The context is a clinical workflow where multiple agents are responsible for different tasks.
16 Your job is to ensure that data from one step is passed correctly to the next. If any agent's
17 output is incomplete, decide whether to repeat the step or move forward.
18
19 C (Constraints):
20 - Do not perform the tasks yourself—delegate them to the appropriate agent.
21 - Pass the output of one agent as input to the next.
22 - Logically determine the next step in the process based on the progress of the workflow.
23
24 E (Evaluation):
25 Your output is successful if:
26 - The History, Assessment, Treatment, and Medication steps are completed in sequence.
27 - Outputs from each agent are consistent, coherent, and passed to the next agent without loss of context.
28 - The final medication list is relevant and accurate.
29 """,
30     model="gpt-4o-mini",
31     functions=[
32         handoff_to_history_agent,
33         handoff_to_assessment_agent,
34         handoff_to_treatment_agent
35     ] # handoff_to_medication_agent
36     # ]
37 )

1 # 3. Sample Workflow Execution
2 def run_orchestrator_workflow(client, orchestrator_agent,
3                               history_taking_agent,
```



```

4             assessment_agent,
5             treatment_agent):
6         """Executes the Orchestrator Agent workflow."""
7         print("Starting Orchestrator Workflow...\n")
8
9         # Step 1: History Taking Agent
10        gathered_data = handoff_to_history_agent()
11        print("History Taking Completed.\n")
12
13        # Step 2: Assessment Agent
14        assessment_output = handoff_to_assessment_agent(assessment_agent, gathered_data)
15        print("Assessment Completed.\n", assessment_output)
16
17        # Step 3: Treatment Agent
18        treatment_output = handoff_to_treatment_agent(treatment_agent, assessment_output)
19        print("Treatment Plan Generated.\n", treatment_output)
20
21        # # Step 4: Medication List Agent
22        # medication_output = handoff_to_medication_agent(client, medication_list_agent, treatment_output)
23        # print("Medication List Generated.\n", medication_output)
24
25        # Final Output
26        print("\nWorkflow Completed.")
27        # print("Medication List:\n", medication_output)
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
```

```
7     history_taking_agent=history_taking_agent,  
8     assessment_agent=assessment_agent,  
9     treatment_agent=treatment_agent,  
10    medication_agent=medication_agent,  
11    medication_list_agent_query=medication_list_agent_query  
12 )
```

```

Loaded medication list...
Vector store created...
OpenAI model initialized...
Create QA chain...
Orchestrator: Starting the full medical agent workflow...

Orchestrator: Running History Taking Agent...
How are you feeling today? Can you describe any symptoms or sickness you are experiencing?
Your response: dizzy
When did you first start feeling dizzy?
Your response: yesterday
Where do you feel the dizziness—does it feel like you're spinning, lightheaded, or unsteady?
Your response: spinning
How long do the dizzy spells last when they occur?
Your response: 4 hr
How would you describe the dizziness? Is it more like a spinning sensation, lightheadedness, or something else?
Your response: 80 degrees C
What seems to make the spinning sensation worse?
Your response: lying down
What helps alleviate the dizziness, if anything?
Your response: clising my eyes
Does the dizziness occur all the time, or does it come and go?
Your response: all the teim
On a scale from 1 to 10, how severe is the spinning sensation you are experiencing?
Your response: 9
Can you tell me if the dizziness is constant throughout the day, or does it vary with certain activities?
Your response: constant
OLDCART history completed!

```

Patient History (OLDCART + Severity and Temporality):

```

Onset: yesterday
Location: spinning
Duration: 4 hr
Character: 80 degrees C
Aggravating factors: lying down
Relieving factors: clising my eyes
Severity: 9
Temporality <constant | intermittent>: constant
Timing: all the teim

```

Conversation history:

```

User: dizzy
User: yesterday
User: spinning
User: 4 hr
User: 80 degrees C
User: lying down
User: clising my eyes
User: all the teim
User: 9
User: constant

```

```

Assistant: {
  "onset": "yesterday",
  "location": "spinning",
  "duration": "4 hr",
  "character": "80 degrees C",
  "aggravating factors": "lying down",
  "relieving factors": "clising my eyes",
  "severity": "9",
  "temporality": "constant",
  "timing": "all the teim"
}

```

History Taking Agent Output:

```

{
  "onset": "yesterday",
  "location": "spinning",
  "duration": "4 hr",
  "character": "80 degrees C",
  "aggravating factors": "lying down",
  "relieving factors": "clising my eyes",
  "severity": "9",
  "temporality": "constant",
  "timing": "all the teim"
}

```

Orchestrator: Running Assessment Agent...

Assessment Agent Output:

Based on the patient history you've provided, here is a prioritized list of differential diagnoses:

- **Benign Paroxysmal Positional Vertigo (BPPV)**:**
 - The spinning sensation, particularly aggravated by lying down, suggests BPPV, which is characterized by brief episodes of vertigo.
- **Vestibular Neuritis**:**

- The sudden onset of vertigo could indicate vestibular neuritis, which presents with persistent vertigo. The patient

3. ****Meniere's Disease****:

- This condition typically presents with episodic vertigo, tinnitus, and hearing changes. However, the constant nature

4. ****Migrainous Vertigo (Vestibular Migraine)****:

- If the patient has a history of migraines, vestibular migraine could present as recurrent episodes of vertigo, some

5. ****Labyrinthitis****:

- This would also explain acute onset spinning and could be due to a viral infection. However, additional signs of hea

Additional information that would be helpful includes the patient's medical history, any associated symptoms (e.g., hear

Orchestrator: Running Treatment Agent...

Treatment Agent Output:

Based on the differential diagnoses you've outlined and the patient's symptoms of vertigo, here are the recommended next

1. ****Patient History and Physical Exam****:

- Gather a more detailed history, focusing on the character and duration of the vertigo episodes, any associated sympt
- Conduct a physical examination, including a neurological exam, as well as the Dix-Hallpike maneuver to assess for B

2. ****Diagnostic Testing****:

- ****Electronystagmography (ENG) or Videonystagmography (VNG)****: Consider these tests to evaluate vestibular function a
- ****Audiometric Testing****: This can help assess for any hearing deficits that may point toward Meniere's disease or la

3. ****Medication****:

- If BPPV is suspected, reassure the patient and advise them to perform the Epley maneuver, which can help alleviate s
- For vestibular neuritis or if the patient is experiencing significant nausea, prescribe vestibular suppressants (suc

4. ****Referral****:

- If vestibular neuritis, migraine-related vertigo, or labyrinthitis is considered, a referral to a neurologist or oto
- If Meniere's disease is suspected, refer to an ENT specialist for further assessment.

5. ****Lifestyle Modifications****:

- Advise the patient to track their symptoms, including potential triggers and any associated lifestyle factors, and t

6. ****Follow-Up****:

- Schedule a follow-up appointment to reassess the patient's symptoms and response to any initial management strategie

By taking these steps, you will gather the necessary information to differentiate between the possible causes of the pat

Orchestrator: Running Medication List Agent...

Orchestrator: Passing treatment recommendations to Medication Agent...

Orchestrator: Passing treatment recommendations to Medication Agent...

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-76-3f0d492f10db> in <cell line: 5>()
      3
      4 # Run the orchestrator workflow
----> 5 orchestrator_workflow(
      6     client=client,
      7     history_taking_agent=history_taking_agent,
```

⌵ 4 frames

```
<ipython-input-66-f2d3e5e653f2> in handoff_to_prescription_agent(client, medication_agent, treatment_output,
medication_list_agent_query)
     15
```

```
     16     # Use the RAG-based query function to retrieve medication information
----> 17     rag_query_result = medication_list_agent_query(
     18         f"Based on the following treatment recommendations, identify relevant medications, their "
     19         f"dosages, common side effects, and contraindications:\n{treatment_output}"
```

TypeError: 'str' object is not callable