

ASSIGNMENT 2:
SALES RECOMMENDATION AGENT

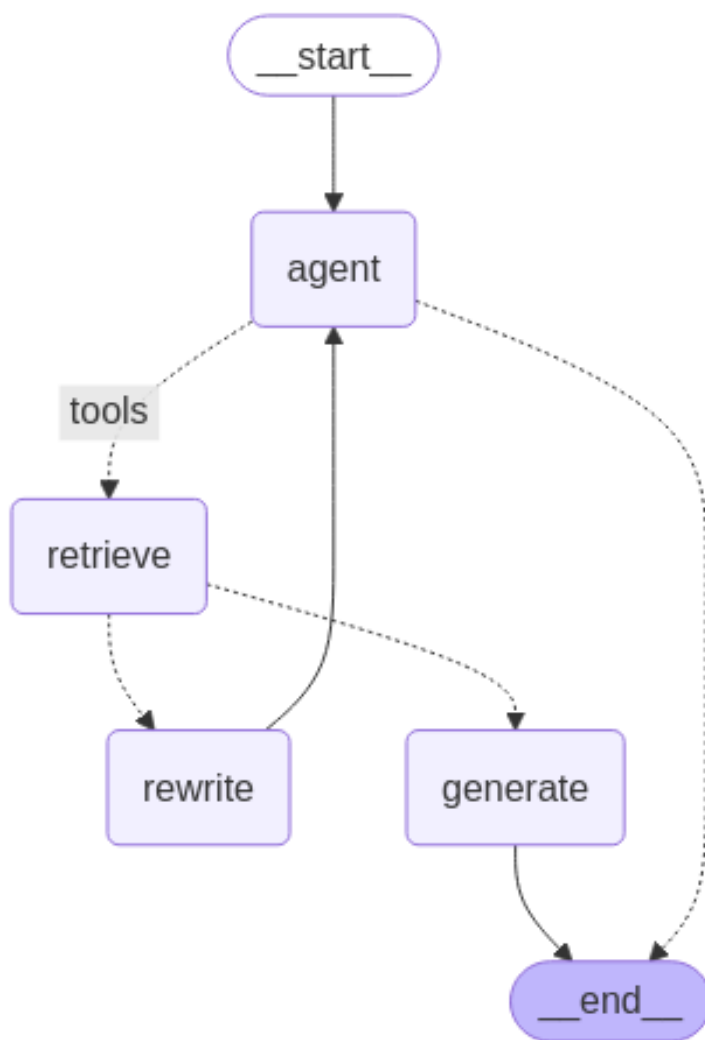
Kevin Geidel
MSDS 442: AI Agent Design & Development
Northwestern University
May 11, 2025

Requirement 1: Graph the Agent with LangChain/LangGraph

The construction of the agent and accompanying graph begins in cell 2 (see appendix). The actual assembly of the graph occurs in cell 5. However, some of the components, such as the retriever **ToolNode** and **AgentState** class, are built above. Following the logic in cell 5 we first instantiate an empty graph:

```
workflow = StateGraph(AgentState)
```

The workflow object has `add_node` and `add_edge` methods that allow us to assemble the components created in cells 2-6. The output is displayed graphically in cell 6 (reproduced below.)



Requirement 2: Load the Dell web pages

Loading the web pages occurs in cell 2. A list of URLs are defined and based to the **WebBaseLoader**. The **RecursiveCharacterTextSplitter** chunks the documents into workable pieces. The result is a list of LangChain documents that each have a portion of each site. The documents are embedded and stored in the ChromaDB vector store.

Requirement 3: Inspect and comment on four queries

The first query sent to the agent was *'I want a dell computer for travel that has Intel® Core™ 7 150U.'* The response is reproduced below. While the response does not mention travel, specifically, it did recommend ultralight designs. Perhaps this was the agent addressing the travel component? These laptops do have Intel 7 processors, so in that sense they are accurate.

```

1 # requirement #3
2 query_agent("I want a dell computer for travel that has Intel® Core™ 7 150U.")
✓ 5.9s Python

---CALL AGENT---
"Output from node 'agent':"
'-----'
{ 'messages': [ AIMessage(content='', additional_kwargs={'tool_calls': [{'index': 0, 'id': 'call_30h30dYUPNUL5tC8o8Pl6zXj', 'function':
---CHECK RELEVANCE---
---DECISION: DOCS RELEVANT---
"Output from node 'retrieve':"
'-----'
{ 'messages': [ ToolMessage(content='16:10 displays, enhanced audio, ultralight option and Intel® Core™ Ultra processor.1. Select bas
---GENERATE---
"Output from node 'generate':"
'-----'
{ 'messages': [ 'For laptops with 16:10 displays, enhanced audio, ultralight '
                  'options, and Intel® Core™ Ultra processors, consider the Dell '
                  'XPS 14 Laptop or the Dell Inspiron 14 2-in-1 Laptop 7440. '
                  'Both models feature Intel processors and are designed for '
                  'portability and performance. The Dell Latitude 5450 is '
                  'another option, though it may vary in specifications.']]

1 # While the response does not mention travel, specifically, it did recommend ultralight designs.
2 # These laptops do have intel 7 processors, so in that sense they are accurate.
✓ 0.0s Python

```

The second query, *'I want a dell computer that has Intel® Core™ Ultra 5 135U vPro® and has 512 GB SSD.'*, and its response are displayed below. It included specific requirements on the desired memory. The response addresses this but also includes other options that have different memory specifications.

```

1 query_agent("I want a dell computer that has Intel® Core™ Ultra 5 135U vPro® and has 512 GB SSD.")
[10] ✓ 5.1s Python
...
---CALL AGENT---
"Output from node 'agent':"
'...'
{ 'messages': [ AIMessage(content='', additional_kwargs={'tool_calls': [{'index': 0, 'id': 'call_yPpFd0ffEfKZY3EHl2uxll0q', 'function': {'arguments': '{"query":"Dell laptop with Intel Core Ultra 5 135U vPro, 16 GB DDR5, and a 512 GB SSD"}'}}]})
---CHECK RELEVANCE---
---DECISION: DOCS RELEVANT---
"Output from node 'retrieve':"
'...'
{ 'messages': [ ToolMessage(content='Core™ Ultra 5 135U, vPro®Windows 11 ProIntel® Graphics16 GB DDR5256 GB SSD14" Non-Touch FHD Tech Specs Select Selected Configuration', tool_call_id='call_yPpFd0ffEfKZY3EHl2uxll0q')
---GENERATE---
"Output from node 'generate':"
'...'
{ 'messages': [ 'For your specifications, I recommend the Dell laptop with an '
                'Intel® Core™ Ultra 5 135U, vPro®, 16 GB DDR5, and a 256 GB '
                'SSD priced at $1,399.00. Alternatively, you could consider '
                'the model with the same processor and memory but a 512 GB SSD '
                'for $1,559.00. Both options feature a 14" Non-Touch FHD '
                'display and run on Windows 11 Pro.']}

1 # The response includes laptops that match the request
2 # but also offer additional models with other storage amounts.
[11] ✓ 0.0s Python

```

The third query, *'I want a dell computer that has Intel® Core™ Ultra 7 165U vPro® and 1 TB SSD'*, takes a similar approach- requesting a particular processor with a particular amount of storage. The agent performs poorly on this query. The recommended laptop was wrong on both counts. The agent is not ensuring the requested specifications are being honored.

```

1 query_agent('I want a dell computer that has Intel® Core™ Ultra 7 165U vPro® and 1 TB SSD?')
2
[12] ✓ 4.3s Python
...
---CALL AGENT---
"Output from node 'agent':"
'...'
{ 'messages': [ AIMessage(content='', additional_kwargs={'tool_calls': [{'index': 0, 'id': 'call_2eTsfpcity0D6Fwa0psGU48c', 'function': {'arguments': '{"query":"Dell laptop with Intel Core Ultra 7 165U vPro, 16 GB DDR5, and a 1 TB SSD"}'}}]})
---CHECK RELEVANCE---
---DECISION: DOCS RELEVANT---
"Output from node 'retrieve':"
'...'
{ 'messages': [ ToolMessage(content='Core™ Ultra 5 135U, vPro®Windows 11 ProIntel® Graphics16 GB DDR5256 GB SSD14" Non-Touch FHD Tech Specs Select Selected Configuration', tool_call_id='call_2eTsfpcity0D6Fwa0psGU48c')
---GENERATE---
"Output from node 'generate':"
'...'
{ 'messages': [ 'For your specifications, I recommend the Dell laptops with '
                'the Intel® Core™ Ultra 5 135U, vPro®, and 16 GB RAM. You can '
                'choose between configurations with either a 256 GB SSD priced '
                'at $1,399.00 or a 512 GB SSD priced at $1,659.00. Both '
                'options feature a 14" Non-Touch FHD display.']}

1 # This response did not retrieve the proper processor.
2 # There are very specific recommendations but they are not matching the requested specs
[13] ✓ 0.0s Python

```

The final query is *'I want a light weight XPS computer with Intel® Core™ Ultra 7 165U vPro® and 1 TB SSD.'* It is very similar to the prior query however the agent handled it better. The recommended models are all 'ultralight.'

```

1 query_agent('I want a light weight XPS computer with Intel® Core™ Ultra 7 165U vPro® and 1 TB SSD.')
14] ✓ 4.2s Python
---CALL AGENT---
"Output from node 'agent':"
'...'
{ 'messages': [ AIMessage(content='', additional_kwargs={'tool_calls': [{'index': 0, 'id': 'call_4fmN5yufGTNWD7nVT3jZX8F', 'function': {'arguments': '{"query":"lightwe
---CHECK RELEVANCE---
---DECISION: DOCS RELEVANT---
"Output from node 'retrieve':"
'...'
{ 'messages': [ ToolMessage(content='16:10 displays, enhanced audio, ultralight option and Intel® Core™ Ultra processor.1. Select base configuration14" 745514" 74502. 9
---GENERATE---
"Output from node 'generate':"
'...'
{ 'messages': [ 'For your specifications, I recommend the Dell 14" 7455 with '
'an Intel® Core™ Ultra 5 135U processor, 16 GB LPDDR5X memory, '
'and a 256 GB SSD, priced at $1,659. Alternatively, you could '
'consider the Dell 14" 7450 with similar specs but a 512 GB '
'SSD for $1,399. Both options feature a 14" Non-Touch FHD '
'display and enhanced audio.')]
<
1 # The response does not mention it specifically but it did recommend a lightweight model.
2 # The memory does not match the requested specs.
15] ✓ 0.0s Python

```

In my experimentation I noticed the agent does better when the prompt is phrased as a question. To test this hypothesis I ran one additional query: *‘What laptop has the Intel 7 processor?’* The agent handles this one perfectly with a complete response that accurately and coherently addresses the prompt.

```

1 query_agent("What laptop has the intel 7 processor?")
16] ✓ 5.2s Python
---CALL AGENT---
"Output from node 'agent':"
'...'
{ 'messages': [ AIMessage(content='', additional_kwargs={'tool_calls': [{'index': 0, 'id': 'call_32yr7lgayC2WAWhd3w1JYM2y', 'function': {'arguments': '{"query":"laptop
---CHECK RELEVANCE---
---DECISION: DOCS RELEVANT---
"Output from node 'retrieve':"
'...'
{ 'messages': [ ToolMessage(content='Dell Inspiron 14 2 in 1 Laptop 7440 with Intel processor | Dell\n\nlaptop with multiple modes, built with Intel Core processors and
---GENERATE---
"Output from node 'generate':"
'...'
{ 'messages': [ 'For a versatile laptop, consider the Dell Inspiron 14 2-in-1 '
'Laptop 7440, which features a 13th Gen Intel® Core™ i5 '
'processor, 16GB LPDDR5x memory, and a 14" FHD+ touch display. '
'It is designed for multiple modes and is recommended with '
'Windows 11 Pro for business use. The price is approximately '
'$499.99, offering good value for its specifications.')]

```

Requirement 4: Response consistency

The experiments conducted in the assignment show that responses from the agent lack consistency. This is curious when we note the low (0.2) temperature we use when invoking the LLM. Running the same query over and over provides varied results. Sometimes the agent fails to see the question as relevant, sometimes the agent provides a very inaccurate response and sometimes the agent provides a suitable answer that addresses the prompt (directly or indirectly) with varying amounts of extraneous information. This is a focus for the proposed improvements to the agent listed below.

Requirement 5: Agent improvements

There are two primary issues that must be addressed. The first, is the agent failing to recommend laptops that meet the specifications in the prompt. Sometimes they are ignored, other times the recommendations are not limited to those specs. To improve this we can construct another node- recommendation vetting. This step can have the LLM evaluate the proposed response against the original human message. If the response does not meet the test the agent must circle back around and try again.

The second flaw that needs to be addressed is the lack of consistency. This might be mitigated by altering the way we preprocess the documents that are available for inclusion in the context. It may be advantageous to extract the specifications from the documents and store these along side the raw source material. Our experiment has suggested the agent performs well when simply listing the specs of a given model. The LLM could be leveraged to distill the components of each available model and, in turn, provide a clear manifest for subsequent queries.