

Appendix B - U.S. Patent No. 7,398,209

Cortana

Element	Analysis for Cortana
[1.0]. A method responsive to a user generated natural language speech utterance, comprising:	Cortana meets this element.  https://www.youtube.com/watch?app=desktop&v=xuiKfUgEvfo

Element	Analysis for Cortana
	 The screenshot shows the Windows 10 Settings application. On the left is a navigation pane with icons for Settings, Home, Camera, Location, and Speech Recognition. The main pane is titled 'Settings' and contains the following text: 'Cortana can give you suggestions, ideas, reminders, alerts and more.' Below this is a toggle switch labeled 'On'. Further down, it says 'Turning Cortana off clears what Cortana knows on this device, but won't delete anything from the Notebook. After Cortana is off, you can decide what you'd like to do with anything still stored in the cloud.' Below that is a link 'Manage what Cortana knows about me in the cloud'. A white rectangular box highlights the 'Hey Cortana' section, which includes the text 'Hey Cortana', 'Let Cortana respond to "Hey Cortana."', a toggle switch labeled 'Off', and the note 'Cortana is always ready when this is on, which uses more battery.' At the bottom of the settings pane, it says 'Find flights and more' and 'Detect tracking info, such as flights, in messages'. The taskbar at the bottom shows the Windows logo, 'Ask me anything' search bar, and icons for Chrome, File Explorer, and Photoshop. <i>Use the Cortana Voice Assistant in Windows 10, July 15, 2015,</i> https://windows.gadgethacks.com/how-to/use-cortana-voice-assistant-windows-10-0162955/
[1.1] receiving the user generated natural language speech utterance, the received user utterance	Cortana meets this element. See element [1.0].

Element	Analysis for Cortana
containing at least one request;	
[1.2] maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance;	Cortana meets this element. For example, it uses machine learning and artificial intelligence (AI) technology to understand user intent and references. On information and belief, such technology entails the use of prior probabilities and fuzzy possibilities. Cortana's design philosophy is entrenched in state-of-the-art machine-learning and data-mining algorithms. Both developers and researchers are able to use Microsoft's broad assets across commercial and enterprise products, including strong ties to Bing Web search and Microsoft speech algorithms and data. https://blogs.microsoft.com/blog/2014/04/18/go-behind-the-scenes-of-cortana-with-microsoft-research/ How Cortana learns In the initial stages, when you set up Cortana for your Microsoft account, it gathers your personal information such as name, email, age, and other facts that you provided for that account. Logging into your Microsoft account is needed to use Cortana. Secondly, the more you use Cortana, the better it performs. As stated earlier, Cortana learns and fine tunes itself when you tend to ask more and more questions. Also, allowing it to access your location, search history, photos, documents, music, emails, etc., gives it a chance to optimize its learning. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-cortana/

Element	Analysis for Cortana
	Our PDA LU is built to handle multi-domain, multi-turn, contextual query understanding [14, 9, 15, 16], subject to the constraints of the back-end data sources and experiences. The LU module semantically parses and analyses the queries according to a domain-specific semantic schema. Given a requirement to minimize latency and computational costs (dimensions 7,8) deep semantic analysis is not performed. Instead, LU delivers a flat analysis matching the experience requirements while allowing for free form natural language expressions that represent different user intents and slots. Follow-up queries in isolation are ambiguous, <i>e.g.</i>, • Turn 1: “how is the weather in New York” (weather) • Turn 2: “what about the weekend” (weather) or • Turn 1: “how is my schedule” (calendar) • Turn 2: “what about the weekend” (calendar) LU modeling thus must be performed in a contextual manner. Here, conversational session information from the history greatly reduces the ambiguity of the current turn. We built a set of models addressing contextual domain, intent and slot modeling [14, 15, 16]. Our current approach involves modeling contextual intent prediction as a sequential tagging problem with previous utterances as features that condition this turn’s prediction. Contextual information from beyond LU, such as dialog state (where this is tracked by an experience), and PDA responses are also exploited as effective external features. Contextual modeling of queries reduces the likelihood of abrupt intent or domains switching leading to more coherent interaction during a multi-turn session.

Element	Analysis for Cortana
	5.4. Hypothesis Ranking and Selection HRS is the mechanism by which the platform ranks and selects the final response shown to the user. The processing is done in two stages, with an initial ranking of the hypotheses (HR) followed by selection of the best hypothesis according to rank score and other features (HS). One of the key decisions in dealing with open domain requests is the recognition of requests that are not handled by any specialized experience provider and letting such requests fall through to web search. 4.1. Domain, Intent, Slot Modeling Prior work has shown a variety of approaches can be used for language understanding [17, 8, 20, 21, 15]. Some early systems used rule based parsing, hand-authored by an expert for a specific domain [17]. Recent state-of-the-art systems use machine learned models [21, 20, 15]. Semantic meaning of a query is represented as a semantic frame (SF), which is a tuple of (DOMAIN, INTENT, SLOTS) in the semantic space. The slots are a list of key-value pairs: (SLOTNAME, SLOTVALUE). In our system, domain classification is done using machine learned (ML) classifiers such as SVMs [25], which generate a score for each domain. We use a separate binary one-against-all classifier for each domain. For each domain, we also train a domain specific intent and slot model. Intents and slots can be shared across different domains. Intent detection also uses SVM classification but is framed as a multi-class classification problem. Slot tagging is considered as a sequence classification problem. Conditional random fields (CRFs) [26] and more recently deep learning techniques [9, 31, 16] are used for slot tagging. All of the LU models use weighted entity lexicons [18, 19] and are additionally trained with artificial out-of-vocabulary (OOV) words to improve generalization to unseen entities. The slot values can be further resolved into entities (e.g. a strongly typed object in some back-end data source) or canonicalized into a standard form (e.g. time/date values). To deal with the open domain nature of queries, each LU domain uses training data from all other domains as negative examples, including representative data for the web domain

Element	Analysis for Cortana
	AN OVERVIEW OF END-TO-END LANGUAGE UNDERSTANDING AND DIALOG MANAGEMENT FOR PERSONAL DIGITAL ASSISTANTS, Microsoft Corporation, 2016, available at https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/CortanaLUDialog-FromSLTproceedings.pdf (emphasis added)
[1.3]. recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;	Cortana meets this element. For example, on information and belief, Cortana recognizes words and phrases using one or more data structures that stores information relating to a user, including the user’s preferences, location, and conversation history. How Cortana learns In the initial stages, when you set up Cortana for your Microsoft account, it gathers your personal information such as name, email, age, and other facts that you provided for that account. Logging into your Microsoft account is needed to use Cortana. Secondly, the more you use Cortana, the better it performs. As stated earlier, Cortana learns and fine tunes itself when you tend to ask more and more questions. Also, allowing it to access your location, search history, photos, documents, music, emails, etc., gives it a chance to optimize its learning. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-cortana/ In addition, on information and belief, Cortana also uses machine learning and artificial intelligence (AI) technology to recognize words and phrases, and such technology is trained on dictionary and phrase tables.

Element	Analysis for Cortana
	4. LANGUAGE UNDERSTANDING The language understanding (LU) component uses as input either typed text or speech transcription from an open vocabulary speech recognition service and performs semantic analysis on the query to determine the underlying user's intent [8, 9, 4, 15]. Our PDA LU is built to handle multi-domain, multi-turn, contextual query understanding [14, 9, 15, 16], subject to the constraints of the back-end data sources and experiences. The LU module semantically parses and analyses the queries according to a domain-specific semantic schema. Given a requirement to minimize latency and computational costs (dimensions 7,8) deep semantic analysis is not performed. Instead, LU delivers a flat analysis matching the experience requirements while allowing for free form natural language expressions that represent different user intents and slots. deep learning techniques [9, 31, 16] are used for slot tagging. All of the LU models use weighted entity lexicons [18, 19] and are additionally trained with artificial out-of-vocabulary (OOV) words to improve generalization to unseen entities. The slot values can be further resolved into entities (<i>e.g.</i> a strongly typed object in some back-end data source) or canonicalized into a standard form (<i>e.g.</i> time/date values). To deal with the open domain nature of queries, each LU domain uses training data from all other domains as negative examples, including representative data for the web domain AN OVERVIEW OF END-TO-END LANGUAGE UNDERSTANDING AND DIALOG MANAGEMENT FOR PERSONAL DIGITAL ASSISTANTS, Microsoft Corporation, 2016, available at https://www.microsoft.com/en-us/research/wp-

Element	Analysis for Cortana
	content/uploads/2016/12/CortanaLUDialog-FromSLTproceedings.pdf (emphasis added)
[1.4] parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases;	Cortana meets this element. For example, Cortana uses keywords to parse user queries. 5.1. Slot/Entity Carry Over (SCO) and Co-reference Tracking keywords contained in slots and associated entities enables a certain set of user initiative experiences, <i>e.g.</i> • Turn 1: “find french restaurants in seattle” • State 1: cuisine=“french”, place_type=“restaurants”, absolute_location=“seattle” • Turn 2: “how about chinese” • State 2: cuisine=“chinese”, place_type=“restaurants”, absolute_location=“seattle” AN OVERVIEW OF END-TO-END LANGUAGE UNDERSTANDING AND DIALOG MANAGEMENT FOR PERSONAL DIGITAL ASSISTANTS, Microsoft Corporation, 2016, available at https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/CortanaLUDialog-FromSLTproceedings.pdf For example, on information and belief, Cortana uses natural language processing (NLP) and natural language understanding (NLU) technology to determine a context for a user utterance based on one or more key words. For example, Cortana uses the keyword “weather” to determine a weather-related context of a request.

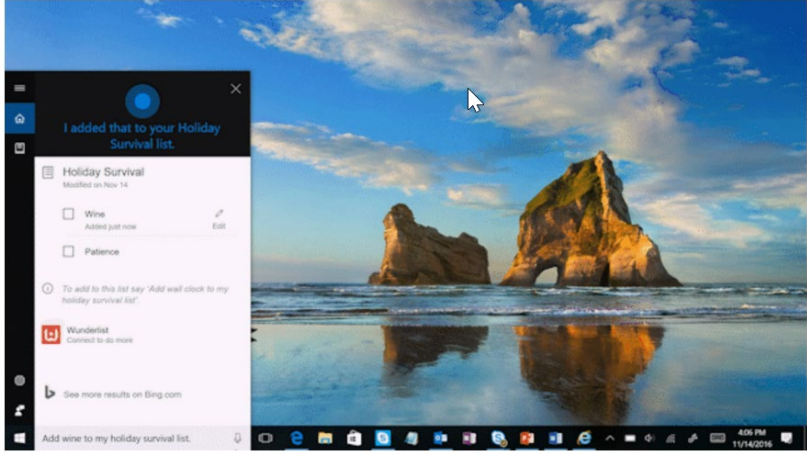
Element	Analysis for Cortana
	Our PDA LU is built to handle multi-domain, multi-turn, contextual query understanding [14, 9, 15, 16], subject to the constraints of the back-end data sources and experiences. The LU module semantically parses and analyses the queries according to a domain-specific semantic schema. Given a requirement to minimize latency and computational costs (dimensions 7,8) deep semantic analysis is not performed. Instead, LU delivers a flat analysis matching the experience requirements while allowing for free form natural language expressions that represent different user intents and slots. Follow-up queries in isolation are ambiguous, <i>e.g.</i>, • Turn 1: “how is the weather in New York” (weather) • Turn 2: “what about the weekend” (weather) or • Turn 1: “how is my schedule” (calendar) • Turn 2: “what about the weekend” (calendar) LU modeling thus must be performed in a contextual manner. Here, conversational session information from the history greatly reduces the ambiguity of the current turn. We built a set of models addressing contextual domain, intent and slot modeling [14, 15, 16]. Our current approach involves modeling contextual intent prediction as a sequential tagging problem with previous utterances as features that condition this turn’s prediction. Contextual information from beyond LU, such as dialog state (where this is tracked by an experience), and PDA responses are also exploited as effective external features. Contextual modeling of queries reduces the likelihood of abrupt intent or domains switching leading to more coherent interaction during a multi-turn session.

Element	Analysis for Cortana
	5.4. Hypothesis Ranking and Selection HRS is the mechanism by which the platform ranks and selects the final response shown to the user. The processing is done in two stages, with an initial ranking of the hypotheses (HR) followed by selection of the best hypothesis according to rank score and other features (HS). One of the key decisions in dealing with open domain requests is the recognition of requests that are not handled by any specialized experience provider and letting such requests fall through to web search. AN OVERVIEW OF END-TO-END LANGUAGE UNDERSTANDING AND DIALOG MANAGEMENT FOR PERSONAL DIGITAL ASSISTANTS, Microsoft Corporation, 2016, available at https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/CortanaLUDialog-FromSLTproceedings.pdf (emphasis added) Microsoft's Cortana Intelligence Institute cofounder on building better AI News By Jason Ward last updated August 12, 2020 Why is understanding context is so important? A lot of the help that we require from digital assistants like Cortana depends strongly on the current context. Even location and time of day are important signals about the types of assistance that people are likely to require. For example, we see that users are more likely to request sleep sounds at home in the evening and productivity support, such as calendaring or email, at work during the day. Richer information about the current situation enables better support from Cortana. https://www.windowscentral.com/cortana-intelligence-institute-co-founder-ryen-white-talks-cortana

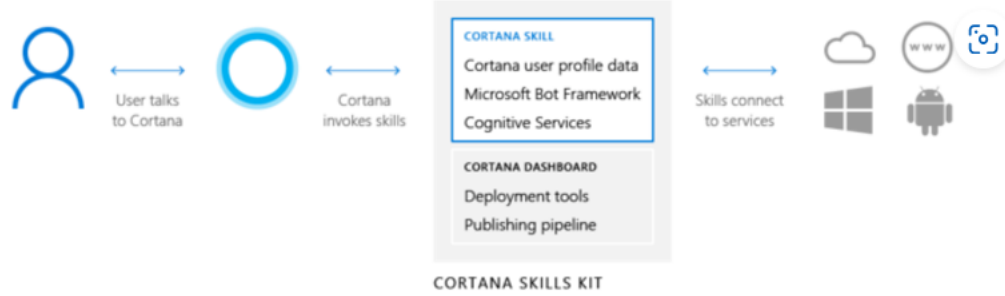
Element	Analysis for Cortana
[1.5] selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context;	Cortana meets this element. On information and belief, Cortana uses multiple domain agents, and each domain agent is specialized in processing requests in its area of expertise. Top 5 Features of Cortana Cortana’s ability to perform several tasks for its users, right from handling user queries, calendar events to managing smart home devices, makes it an appropriate multifunctional tool that anyone can use to ease their lives. It gives tough competition to Alexa and Siri when it comes to handling such a variety of functions. Let’s take a look at the top five key features of Cortana: 1. Create reminders Cortana is a handy digital assistant that helps in creating reminders without much intervention. For example, you can address Cortana by saying, ‘Hey Cortana, remind me to attend Tom’s conference call at 6 p.m’. You can also enable location settings and give a command to Cortana, such as, ‘Hey Cortana, remind me to buy groceries when I am returning home from the office in the evening’. Another way of creating reminders is by doing so manually. You can follow these steps: Open up Cortana > click on the Notebook icon > Tap Reminder on the left menu > create a reminder.

Element	Analysis for Cortana
	2. Control smart home devices Cortana allows you to control several home devices such as Philips Hue Lights just by giving voice commands. For example, once you connect the Philips lighting system to Cortana, you can manage it with your voice. You can give instructions such as, ‘Hey Cortana, please dim the lights after 15 minutes’ or ‘Hey Cortana, please switch on the lights in the living room’. 3. Look up words while browsing Cortana provides features to look up words or phrases while browsing the web. The assistant is already enabled on the Microsoft Edge browser, and makes it easy for users to search on the web. This feature can be availed by highlighting the words or phrases in contention, right-clicking on them, and then tapping on the ‘Ask Cortana’ option. This lets Cortana use the Bing search engine to identify the relevant information on the web and then show the results on the browser via a tiny pop up or window.

Element	Analysis for Cortana
	4. Get weather forecasts One of the easiest ways of checking weather forecasts on Cortana-enabled devices is by asking Cortana to report the weather conditions. You can give a voice command such as, ‘Hey Cortana, what’s the current humidity level in Los Angeles’ or ‘Hey Cortana, what will the outside temperature be in my city at 2 p.m.’. Such queries will allow Cortana to look for weather reports pertaining to the said location and also display the identified results for you. 5. Ask for directions While searching for locations on maps, it is quite inconvenient to input or type-in the entire address. Such inconvenience is resolved by Cortana as it allows you to input voice commands and performs the tasks of locating, finding the route, and also giving you directions to the destination all by itself. You can simply say, ‘Hey Cortana, how can I reach XYZ (location name)?’ or ‘Hey Cortana, which route should I take to reach XYZ?’. Besides these, Cortana comes with plenty of other features such as playing music using voice, setting calendar events, performing mathematical calculations, and carrying out measurement and currency conversions. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-cortana/

Element	Analysis for Cortana
	Blog / Mobile / Android / Post Cortana Picks Up To-Do List Skills Paul Thurrott NOV 17, 2016 8  Microsoft announced today that it has improved its Cortana personal digital assistant with a new “to do list” skill. https://www.thurrott.com/windows/windows-10/85172/cortana-picks-list-skills In addition, developers can build Cortana skills in different domains.  May 10, 2017 PC Cortana Skills Kit empowers developers to build intelligent experiences for millions of users Gurpreet Singh Pall https://blogs.windows.com/windowsdeveloper/2017/05/10/cortana-skills-kit-empowers-developers-

Element	Analysis for Cortana																						
	build-intelligent-experiences-millions-users/ Table 1. Data discussion <table border="1"> <thead> <tr> <th data-bbox="598 354 693 383">Domain</th><th data-bbox="806 354 1056 383">Supported Experiences</th></tr> </thead> <tbody> <tr> <td data-bbox="598 391 674 420">Alarm</td><td data-bbox="806 391 1182 420">setting, querying, and deleting alarms</td></tr> <tr> <td data-bbox="598 423 705 453">Calendar</td><td data-bbox="806 423 1188 453">creating and querying calendar entries</td></tr> <tr> <td data-bbox="598 456 772 485">Communication</td><td data-bbox="806 456 1260 485">making phone calls, sending emails and SMS</td></tr> <tr> <td data-bbox="598 488 768 518">Device Settings</td><td data-bbox="806 488 1255 518">commands for changing local device settings</td></tr> <tr> <td data-bbox="598 521 716 550">Document</td><td data-bbox="806 521 1205 550">document search on the local filesystem</td></tr> <tr> <td data-bbox="598 553 762 583">Entertainment</td><td data-bbox="806 553 961 583">music playback</td></tr> <tr> <td data-bbox="598 586 663 615">Local</td><td data-bbox="806 586 1255 631">traffic information, driving directions, search for businesses and attractions</td></tr> <tr> <td data-bbox="598 634 711 664">Reminder</td><td data-bbox="806 634 1213 664">setting, querying, and deleting reminders</td></tr> <tr> <td data-bbox="598 667 695 696">Weather</td><td data-bbox="806 667 1010 696">weather information</td></tr> <tr> <td data-bbox="598 699 653 729">Web</td><td data-bbox="806 699 1234 745">general web search, chit-chat, and targeted question-answering not covered above</td></tr> </tbody> </table> AN OVERVIEW OF END-TO-END LANGUAGE UNDERSTANDING AND DIALOG MANAGEMENT FOR PERSONAL DIGITAL ASSISTANTS, Microsoft Corporation, 2016, available at https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/CortanaLUDialog-FromSLTproceedings.pdf On information and belief, Cortana selects a domain agent based on the determined meaning. For example, the question, “Who won today’s football match between Liverpool and Barcelona?” prompts a sports skill that follows the rules of a pre-coded script to fill in slots for the type of sport, information requested, date and teams. “Will it rain this weekend?” prompts a weather skill and follows pre-scripted rules to get the weekend forecast. https://news.microsoft.com/source/features/ai/microsoft-build-future-of-natural-language/	Domain	Supported Experiences	Alarm	setting, querying, and deleting alarms	Calendar	creating and querying calendar entries	Communication	making phone calls, sending emails and SMS	Device Settings	commands for changing local device settings	Document	document search on the local filesystem	Entertainment	music playback	Local	traffic information, driving directions, search for businesses and attractions	Reminder	setting, querying, and deleting reminders	Weather	weather information	Web	general web search, chit-chat, and targeted question-answering not covered above
Domain	Supported Experiences																						
Alarm	setting, querying, and deleting alarms																						
Calendar	creating and querying calendar entries																						
Communication	making phone calls, sending emails and SMS																						
Device Settings	commands for changing local device settings																						
Document	document search on the local filesystem																						
Entertainment	music playback																						
Local	traffic information, driving directions, search for businesses and attractions																						
Reminder	setting, querying, and deleting reminders																						
Weather	weather information																						
Web	general web search, chit-chat, and targeted question-answering not covered above																						

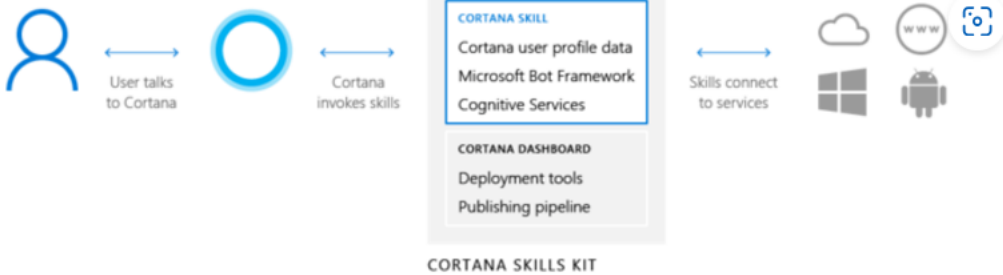
Element	Analysis for Cortana
	How Cortana skills work  How Cortana skills tie into Cognitive Services (click to enlarge) A Cortana skill is essentially a channel of artificial intelligence accessed through Cortana. There are built-in skills like Search, setting reminders, or launching applications; and there are add-in skills built by third parties. Add-in skills are accessed via invocations, key phrases which are registered with Microsoft, so that when the user says, for example, “Ask <invocation Name> <something>”, the question is passed to the registered service rather than being handled by built-in skills. There are currently 17 words you can use before the invocation name, though these are language specific. Unfortunately the only language available in the preview is US English, though if you are in the UK or elsewhere, you can easily set Cortana to US English in order to test a Cortana skill. https://www.theregister.com/2017/07/26/hands_on_with_cortana_skills/

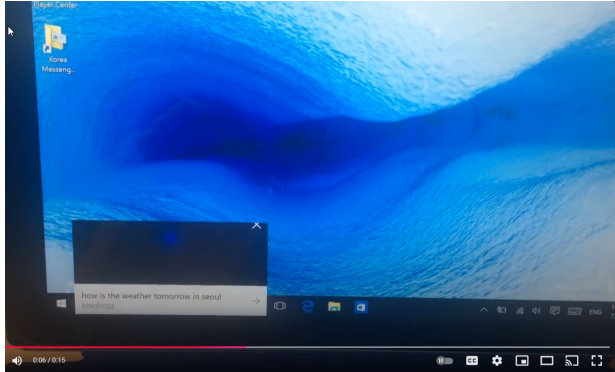
Element	Analysis for Cortana
	Chaining Cortana skills Skills enable users to get things done, but it's rare that completing a single task solves a real-world need. Skill chaining allows a user to use multiple skills in succession to complete a task. Context from each skill is passed to the next skill, ensuring an effortless skill experience. This allows users to work with Cortana to seamlessly move from one task to the next, just as they would with a real human assistant. https://thewindowsupdate.com/2017/12/19/new-features-available-for-cortana-skills-developers/ Microsoft's Cortana Intelligence Institute cofounder on building better AI  By Jason Ward last updated August 12, 2020

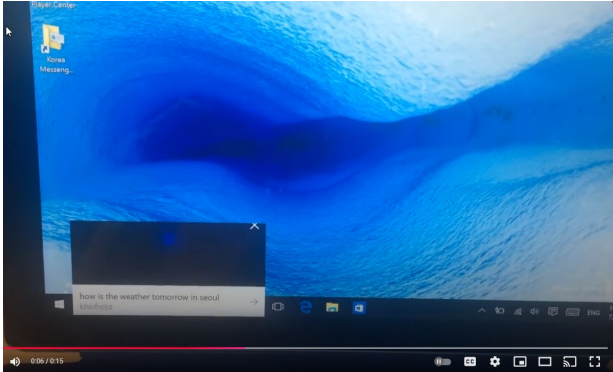
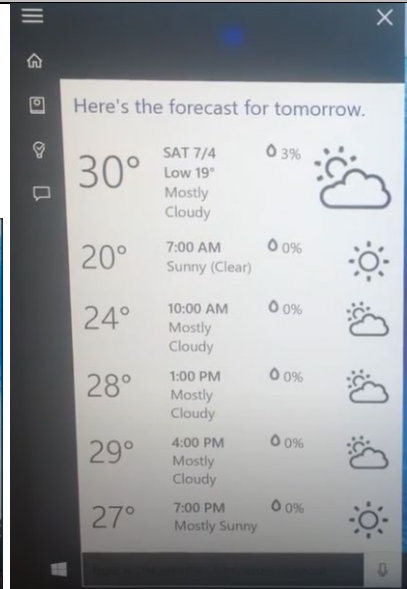
Element	Analysis for Cortana
	Can you give a technical explanation of what has to happen to move from "query-based support", to being able to help with chores to having a rich dialogue with users. as was announced is the institute's goal for Cortana? “ Skill chaining allows multiple Cortana skills in succession so users can move seamlessly from one task to the next. Cortana has a unique, holistic perspective across applications and devices. The key aspect of supporting tasks—from cooking a meal to arranging a meeting—is that they almost always require multiple steps. This is an area where Cortana can engage in a rich dialog with users and make suggestions to them about actions (skills) she can take to help them. In fact, we just announced skill chaining as a way for Cortana skill developers to connect skills to offer more comprehensive task completion support. Cortana Research is exploring contextual skill recommendation, an important enhancement to skill chaining that uses the current context to highlight just what Cortana can do. https://www.windowcentral.com/cortana-intelligence-institute-co-founder-ryen-white-talks-cortana 5.4. Hypothesis Ranking and Selection HRS is the mechanism by which the platform ranks and selects the final response shown to the user. The processing is done in two stages, with an initial ranking of the hypotheses (HR) followed by selection of the best hypothesis according to rank score and other features (HS). One of the key decisions in dealing with open domain requests is the recognition of requests that are not handled by any specialized experience provider and letting such requests fall through to web search. Ranking is carried out over dialog hypotheses, where each dialog hypothesis is an assembly of the SF and an experience provider response (e.g. back-end knowledge plus suggested dialog act). Mak-

Element	Analysis for Cortana
	ing the selection of the system response as late as possible has the benefit that the ML model has full access to all LU analyses, as well as the full session context and knowledge of whether a particular experience produced a response. An experience is allowed to not respond. HRS is contextual in that it should learn to favor the continuation of an existing multi-turn dialog unless the user's utterance indicates a strong likelihood of wishing to switch domains. AN OVERVIEW OF END-TO-END LANGUAGE UNDERSTANDING AND DIALOG MANAGEMENT FOR PERSONAL DIGITAL ASSISTANTS, Microsoft Corporation, 2016, available at https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/CortanaLUDialog-FromSLTproceedings.pdf (emphasis added)
[1.6] formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context;	Cortana meets this element. On information and belief, for example, Cortana formulates a user request according to a grammar that uses slots and entities. 5.1. Slot/Entity Carry Over (SCO) and Co-reference Tracking keywords contained in slots and associated entities enables a certain set of user initiative experiences, <i>e.g.</i>

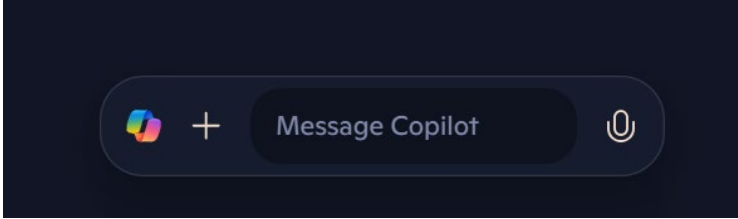
Element	Analysis for Cortana
	• Turn 1: “find french restaurants in seattle” • State 1: cuisine=“french”, place_type=“restaurants”, absolute_location=“seattle” • Turn 2: “how about chinese” • State 2: cuisine=“chinese”, place_type=“restaurants”, absolute_location=“seattle” SCO decides which slots from previous turns are still relevant in the current turn of a multi-turn conversation. It also models implicit start-overs, <i>i.e.</i> dropping of all slots, and operates both within and across domains powering interactions such as “What’s the weather like at Snow Lake?”, “Get me driving directions.” We use a hybrid rule-model approach to implement our SCO system, with the set of rules serving as baseline with a generalized ML model to scale better to new scenarios. Model-based SCO is implemented as a supervised learning task [22]. For each slot from previous turns, the model decides to carry-over or drop slots. Slots in the current turn are unconditionally kept. Our implementation of SCO is as a gradient boosted decision tree, where the concatenation of the current and previous turn utterances are compared to indicate relevant slots. At least one most likely tagging is passed on for further processing. If multiple taggings are produced by SCO, the hypothesis ranking and selection (HRS) component is the ultimate determiner of the winning tagging. For experiences that use strongly typed entities, <i>e.g.</i> celebrity question-answer experiences, an alternative co-reference resolution model exists that learns relationships such as ‘him’ or ‘her’ from mining knowledge bases, <i>e.g.</i> Satori – Microsoft’s equivalent of Freebase or Google’s Knowledge Graph. AN OVERVIEW OF END-TO-END LANGUAGE UNDERSTANDING AND DIALOG MANAGEMENT FOR PERSONAL DIGITAL ASSISTANTS, Microsoft Corporation, 2016, available at https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/CortanaLUDialog-FromSLTproceedings.pdf
[1.7] invoking the selected domain agent to process the formulated request; and	Cortana meets this element. For example, Cortana invokes the selected skill to process the user request.

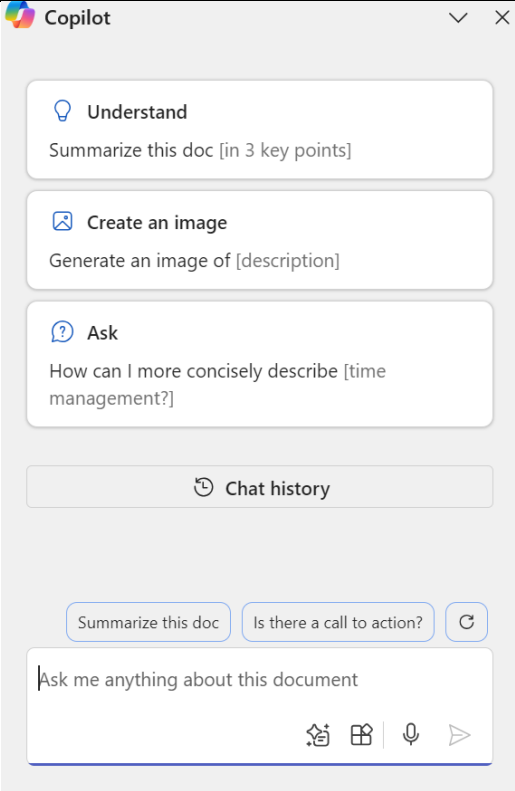
Element	Analysis for Cortana
	How Cortana skills work  How Cortana skills tie into Cognitive Services (click to enlarge) A Cortana skill is essentially a channel of artificial intelligence accessed through Cortana. There are built-in skills like Search, setting reminders, or launching applications; and there are add-in skills built by third parties. Add-in skills are accessed via invocations, key phrases which are registered with Microsoft, so that when the user says, for example, “Ask <invocation Name> <something>”, the question is passed to the registered service rather than being handled by built-in skills. There are currently 17 words you can use before the invocation name, though these are language specific. Unfortunately the only language available in the preview is US English, though if you are in the UK or elsewhere, you can easily set Cortana to US English in order to test a Cortana skill. https://www.theregister.com/2017/07/26/hands_on_with_cortana_skills/ For example, on information and belief, Cortana invokes the selected weather domain agent to process a formulated weather-related request.

Element	Analysis for Cortana
	 https://www.youtube.com/watch?v=b9NCtC_JG4k&ab_channel=Leeyoungsup
[1.8] presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.	Cortana meets this element. For example, Cortana can perform unit conversion, control music, and set reminders, and presents the results to the user using display and/or speech. See Cortana gets some new tricks in Windows 10 (How to), available at https://www.youtube.com/watch?v=7OgETLaQko0&ab_channel=CNET. For example, on information and belief, Cortana presents the results of the processed weather-related request to the user.

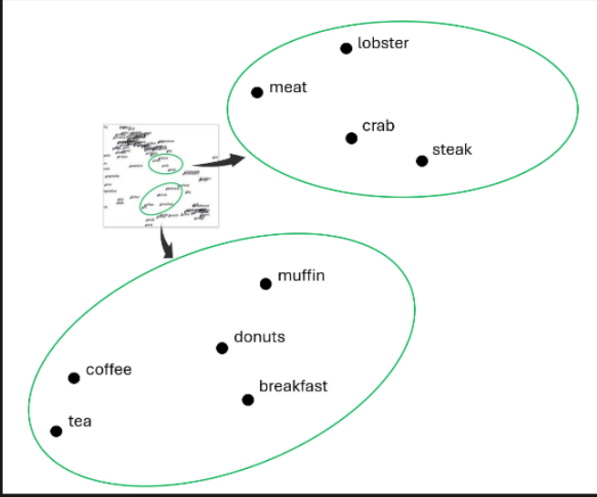
Element	Analysis for Cortana																												
	 <table><tr><th>Time</th><th>Temperature</th><th>Conditions</th><th>Precipitation</th></tr><tr><td>SAT 7/4</td><td>30° / 19°</td><td>Mostly Cloudy</td><td>3%</td></tr><tr><td>7:00 AM</td><td>20°</td><td>Sunny (Clear)</td><td>0%</td></tr><tr><td>10:00 AM</td><td>24°</td><td>Mostly Cloudy</td><td>0%</td></tr><tr><td>1:00 PM</td><td>28°</td><td>Mostly Cloudy</td><td>0%</td></tr><tr><td>4:00 PM</td><td>29°</td><td>Mostly Cloudy</td><td>0%</td></tr><tr><td>7:00 PM</td><td>27°</td><td>Mostly Sunny</td><td>0%</td></tr></table> https://www.youtube.com/watch?v=b9NCtC_JG4k&ab_channel=Leeyoungsup	Time	Temperature	Conditions	Precipitation	SAT 7/4	30° / 19°	Mostly Cloudy	3%	7:00 AM	20°	Sunny (Clear)	0%	10:00 AM	24°	Mostly Cloudy	0%	1:00 PM	28°	Mostly Cloudy	0%	4:00 PM	29°	Mostly Cloudy	0%	7:00 PM	27°	Mostly Sunny	0%
Time	Temperature	Conditions	Precipitation																										
SAT 7/4	30° / 19°	Mostly Cloudy	3%																										
7:00 AM	20°	Sunny (Clear)	0%																										
10:00 AM	24°	Mostly Cloudy	0%																										
1:00 PM	28°	Mostly Cloudy	0%																										
4:00 PM	29°	Mostly Cloudy	0%																										
7:00 PM	27°	Mostly Sunny	0%																										

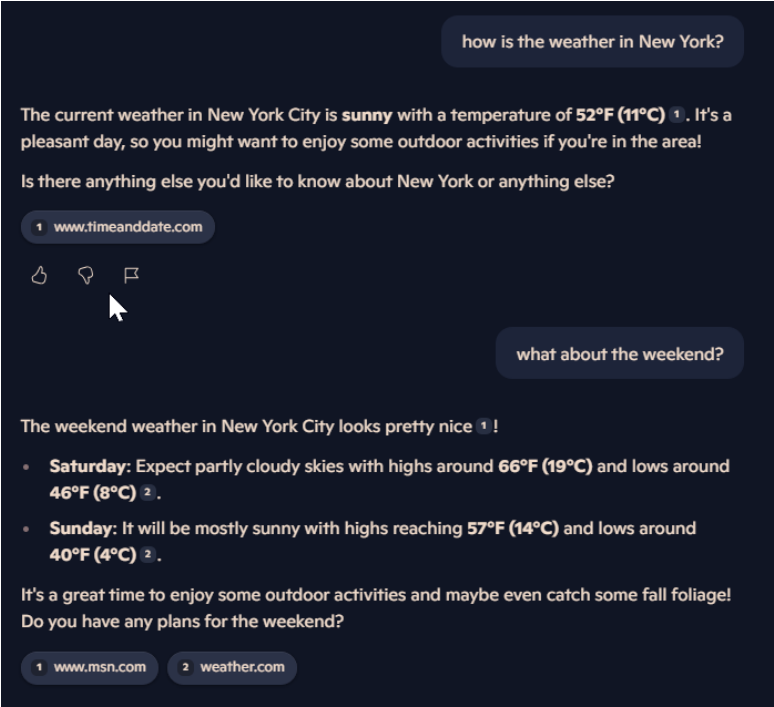
Copilot

Element	Analysis for Copilot
[1.0]. A method responsive to a user generated natural language speech utterance, comprising:	Copilot meets this element. Copilot implements a method responsive to a user generated natural language speech utterance. 

Element	Analysis for Copilot
	
[1.1] receiving the user generated natural language speech utterance, the received user utterance containing at least one request;	Copilot meets this element. See element [1.0].
[1.2]. maintaining a dynamic set of prior probabilities or fuzzy	Copilot meets this element. For example, Copilot uses machine learning and AI technology, which uses probabilities and

Element	Analysis for Copilot
possibilities usable at each stage of processing the received user utterance;	possibilities to understand user intent and references. How does Copilot use AI machine learning? Copilot harnesses the power of AI machine learning to elevate the user experience, enabling the following capabilities: Contextual assistance Copilot employs machine learning to offer contextual assistance that adapts to your needs. Whether you're drafting an email, writing poetry, planning a trip, or researching a topic, Copilot can help suggest relevant information. From brainstorming gift ideas to learning a new skill, use Copilot can help you gather information and resources quickly. For enhanced performance during peak usage, you can upgrade to Copilot Pro so you have assistance when it's most important. Text and images to order Copilot can produce text and images based on your text input, also known as your prompt. Simply enter a descriptive prompt asking Copilot to generate text or an image related to any subject or style. Using machine learning and natural language processing, Copilot will then produce writing or visuals based on your request. Because Copilot is conversational, you can keep asking for tweaks until you receive the output you're after. Predictive text enhancement Machine learning within Copilot enhances its predictive text capabilities, enabling it to anticipate your writing style and intent. Whether you're composing an email or a document, Copilot can help suggest word choices and sentence structures that align with your unique voice, saving you time and ensuring your content is tailored to you. Thanks to machine learning, the more you use Copilot, the better its results. That's one way it's always evolving to enhance efficiency and productivity for users. Try Copilot and the Copilot mobile app today for AI assistance anytime, anywhere. https://www.microsoft.com/en-us/bing/do-more-with-ai/what-is-machine-learning?form=MA13KP For example, Copilot uses semantic index to parse user prompt, which involves using keyword and prior probabilities or fuzzy possibilities.

Element	Analysis for Copilot
	How the semantic index helps manage your data Semantic index enhances the features of Microsoft 365 that allow you to find relevant content based on keywords, personal preferences, and social connections. It does this by creating vectorized indices. A vector is a numerical representation of a word, image pixel, or other data point. The vector is arranged or mapped with close numbers placed in proximity to one another to represent similarity. Vectors are stored in multi-dimensional spaces where semantically similar data points are clustered together in the vector space, enabling Microsoft 365 to handle a broader set of search queries beyond "exact match." In practical terms, this means that Microsoft 365 services such as Microsoft 365 Copilot can: • Understand relationships between different forms of words (for example, tech, technology, technologies; USA, U.S.A, United States, United States of America; dog, cat, pet). • Capture synonyms to expand the amount of searchable information, including the intent of sentences, snippets, documents, and meetings. • Identify related assets to your query or sample content. The following graphic uses text (instead of numbers used by vectorized indices) to show an example of similarity between data points:  https://learn.microsoft.com/en-us/microsoftsearch/semantic-index-for-copilot?WT.mc_id=M365-

Element	Analysis for Copilot
	MVP-5000493
[1.3] recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;	Copilot meets this element. Copilot recognizes words and phrases.  For example, on information and belief, Copilot recognizes words and phrases using information relating to a user, including the user's documents, chats, emails, and contacts.

Element	Analysis for Copilot
	How does Microsoft 365 Copilot use your proprietary organizational data? Microsoft 365 Copilot provides value by connecting LLMs to your organizational data. Microsoft 365 Copilot accesses content and context through Microsoft Graph. It can generate responses anchored in your organizational data, such as user documents, emails, calendar, chats, meetings, and contacts. Microsoft 365 Copilot combines this content with the user's working context, such as the meeting a user is in now, the email exchanges the user had on a topic, or the chat conversations the user had last week. Microsoft 365 Copilot uses this combination of content and context to help provide accurate, relevant, and contextual responses. Data stored about user interactions with Microsoft 365 Copilot When a user interacts with Microsoft 365 Copilot (using apps such as Word, PowerPoint, Excel, OneNote, Loop, or Whiteboard), we store data about these interactions. The stored data includes the user's prompt and Copilot's response, including citations to any information used to ground Copilot's response. We refer to the user's prompt and Copilot's response to that prompt as the "content of interactions" and the record of those interactions is the user's Copilot activity history. For example, this stored data provides users with Copilot activity history in Business Chat and meetings in Microsoft Teams. This data is processed and stored in alignment with contractual commitments with your organization's other content in Microsoft 365. The data is encrypted while it's stored and isn't used to train foundation LLMs, including those used by Microsoft 365 Copilot. To view and manage this stored data, admins can use Content search or Microsoft Purview. Admins can also use Microsoft Purview to set retention policies for the data related to chat interactions with Copilot. For more information, see the following articles: • Overview of Content search • Microsoft Purview data security and compliance protections for generative AI apps • Learn about retention for Copilot For Microsoft Teams chats with Copilot, admins can also use Microsoft Teams Export APIs to view the stored data. https://learn.microsoft.com/en-us/copilot/microsoft-365/microsoft-365-copilot-privacy (emphasis added) Copilot also uses machine learning and artificial intelligence (AI) technology to recognize words and

Element	Analysis for Copilot
	phrases, and such technology is trained on dictionary and phrase tables. How does Copilot use AI machine learning? Copilot harnesses the power of AI machine learning to elevate the user experience, enabling the following capabilities: Contextual assistance Copilot employs machine learning to offer contextual assistance that adapts to your needs. Whether you're drafting an email, writing poetry, planning a trip, or researching a topic, Copilot can help suggest relevant information. From brainstorming gift ideas to learning a new skill, use Copilot can help you gather information and resources quickly. For enhanced performance during peak usage, you can upgrade to Copilot Pro so you have assistance when it's most important. Text and images to order Copilot can produce text and images based on your text input, also known as your prompt. Simply enter a descriptive prompt asking Copilot to generate text or an image related to any subject or style. Using machine learning and natural language processing, Copilot will then produce writing or visuals based on your request. Because Copilot is conversational, you can keep asking for tweaks until you receive the output you're after. Predictive text enhancement Machine learning within Copilot enhances its predictive text capabilities, enabling it to anticipate your writing style and intent. Whether you're composing an email or a document, Copilot can help suggest word choices and sentence structures that align with your unique voice, saving you time and ensuring your content is tailored to you. Thanks to machine learning, the more you use Copilot, the better its results. That's one way it's always evolving to enhance efficiency and productivity for users. Try Copilot and the Copilot mobile app today for AI assistance anytime, anywhere. https://www.microsoft.com/en-us/bing/do-more-with-ai/what-is-machine-learning?form=MA13KP
[1.4] parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in	Copilot meets this element. For example, Copilot uses keywords such as trigger phrases to parse user input and determine its context.

Element	Analysis for Copilot
the recognized words and phrases;	 The screenshot shows a dark-themed article page from Microsoft Copilot Studio. At the top, there is a breadcrumb 'Learn / Microsoft Copilot Studio /'. The main title is 'Optimizing trigger phrases and natural language understanding'. Below the title, it says 'Article • 07/02/2024 • 3 contributors' and a 'Feedback' link. A section titled 'In this article' lists four topics: 'What are trigger phrases in Copilot Studio', 'The importance of the triggering context', 'On punctuation', and 'Creating new trigger phrases', followed by a 'Show 3 more' link. The main content area has a sub-header 'What are trigger phrases in Copilot Studio' and a bulleted list: 'Trigger phrases train your copilot's natural language understanding (NLU) model.', 'Trigger phrases are configured at the topic level and indicate to the copilot for what typical user utterances a specific topic should be triggered.', and 'Trigger phrases typically capture the way an end-user would ask about a problem or issue. For example, "problem with weeds in lawn"'. A mouse cursor is visible over the third bullet point.

Element	Analysis for Copilot
	The importance of the triggering context Copilot Studio NLU behaves differently based on the conversation state, which can sometimes lead to different behaviors for the same user utterance. The following are the different conversation states: • Start of the conversation: the copilot has no context, so a user utterance is expected to either: trigger a topic directly (intent recognition), trigger a "did you mean" (Multiple Topics Matched) disambiguation question among intent candidates if there are multiple matching topics, or go to a fallback topic if the intent isn't recognized. • After a "did you mean" (Multiple Topics Matched) is triggered: NLU optimizes to match one of the suggested topics, with higher thresholds to move out of the presented options. • Switching out from a current topic: If the NLU is trying to fill a slot in a topic, and the user is giving a user query that could trigger another topic (topic switching). https://learn.microsoft.com/en-us/microsoft-copilot-studio/guidance/trigger-phrases-best-practices For example, on information and belief, Copilot uses AI machine learning that determines a context for a user utterance based on one or more key words.

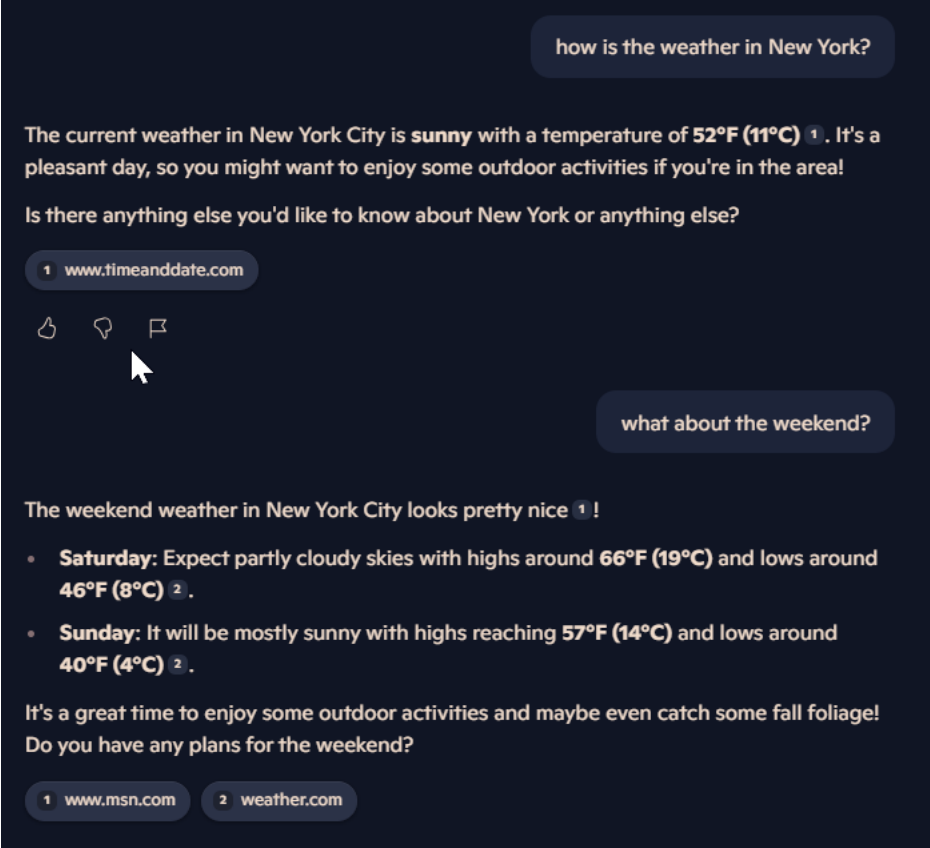
Element	Analysis for Copilot
	How does Copilot use AI machine learning? Copilot harnesses the power of AI machine learning to elevate the user experience, enabling the following capabilities: Contextual assistance Copilot employs machine learning to offer contextual assistance that adapts to your needs. Whether you're drafting an email, writing poetry, planning a trip, or researching a topic, Copilot can help suggest relevant information. From brainstorming gift ideas to learning a new skill, use Copilot can help you gather information and resources quickly. For enhanced performance during peak usage, you can upgrade to Copilot Pro so you have assistance when it's most important. Text and images to order Copilot can produce text and images based on your text input, also known as your prompt. Simply enter a descriptive prompt asking Copilot to generate text or an image related to any subject or style. Using machine learning and natural language processing, Copilot will then produce writing or visuals based on your request. Because Copilot is conversational, you can keep asking for tweaks until you receive the output you're after. Predictive text enhancement Machine learning within Copilot enhances its predictive text capabilities, enabling it to anticipate your writing style and intent. Whether you're composing an email or a document, Copilot can help suggest word choices and sentence structures that align with your unique voice, saving you time and ensuring your content is tailored to you. Thanks to machine learning, the more you use Copilot, the better its results. That's one way it's always evolving to enhance efficiency and productivity for users. Try Copilot and the Copilot mobile app today for AI assistance anytime, anywhere. https://www.microsoft.com/en-us/bing/do-more-with-ai/what-is-machine-learning?form=MA13KP For example, Copilot uses semantic index to parse user prompt, which involves using keyword and prior probabilities or fuzzy possibilities.

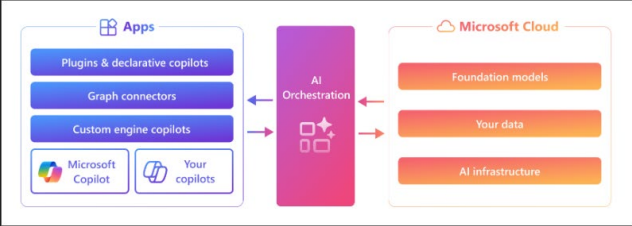
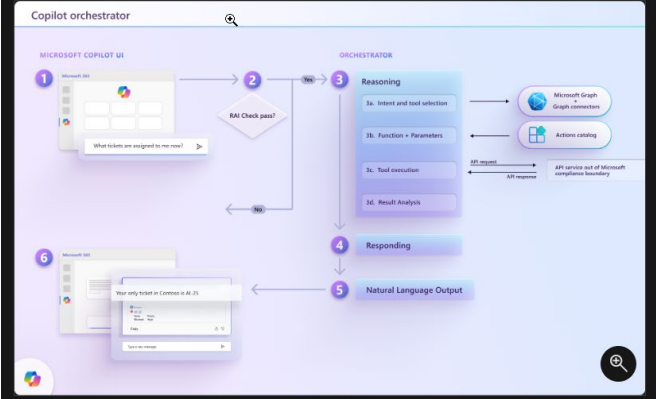
Element	Analysis for Copilot
	How the semantic index helps manage your data Semantic index enhances the features of Microsoft 365 that allow you to find relevant content based on keywords, personal preferences, and social connections. It does this by creating vectorized indices. A vector is a numerical representation of a word, image pixel, or other data point. The vector is arranged or mapped with close numbers placed in proximity to one another to represent similarity. Vectors are stored in multi-dimensional spaces where semantically similar data points are clustered together in the vector space, enabling Microsoft 365 to handle a broader set of search queries beyond "exact match." In practical terms, this means that Microsoft 365 services such as Microsoft 365 Copilot can: • Understand relationships between different forms of words (for example, tech, technology, technologies; USA, U.S.A, United States, United States of America; dog, cat, pet). • Capture synonyms to expand the amount of searchable information, including the intent of sentences, snippets, documents, and meetings. • Identify related assets to your query or sample content. The following graphic uses text (instead of numbers used by vectorized indices) to show an example of similarity between data points: https://learn.microsoft.com/en-us/microsoftsearch/semantic-index-for-copilot?WT.mc_id=M365-

Element	Analysis for Copilot
	<u>MVP-5000493</u> On information and belief, Copilot uses natural language understanding (NLU) technology to determine a context based on keywords and their associated probabilities and possibilities. How does Copilot use NLU? <u>Copilot</u> leverages NLU to enhance user interactions by understanding the context and intent behind user inputs. When a user asks a question or makes a request, Copilot uses NLU to parse the text, identify key entities, and determine the user's intent. This allows Copilot to provide accurate and contextually relevant responses, as in the examples below: 1. Text prompts • Customer service assistant: When a user types, "I'm having trouble with my order," Copilot uses NLU to understand that the user is experiencing an issue and needs assistance, even though the word "help" is not explicitly mentioned. • Virtual assistant: If a user says, "Remind me to call mom tomorrow," Copilot interprets the intent to set a reminder for a specific task and time. • Sentiment analysis: When a user writes a review saying, "The product is amazing, but the delivery was slow," Copilot can identify the positive sentiment towards the product and the negative sentiment towards the delivery. 2. Image prompts • Image captioning: When provided with an image of a dog playing in the park, Copilot can generate a caption like, "A happy dog playing in the park," understanding the context and sentiment of the image. • Object recognition: Given an image of a cluttered desk, Copilot can identify and list objects such as a laptop, coffee mug, and notepad, understanding the context of a workspace. • Scene description: For an image of a beach at sunset, Copilot can describe the scene as, "A serene beach with the sun setting over the horizon," capturing the context and sentiment of the image. <u>https://www.microsoft.com/en-us/bing/do-more-with-ai/what-is-natural-language-understanding?form=MA13KP</u> On information and belief, Copilot determines a context or intent based on its score.

Element	Analysis for Copilot															
	External intents CLU integration supports recognized external intents that utilize the <code>System.Recognizer.IntentOptions</code> system variable in the following scenarios: • Reuse <i>TopicId</i> for the external intent id • Reuse <i>TriggerId</i> for the external intent id • Create a new property <i>IntentId</i> for the external intent id • System variable(s) that support the reason for triggering the recognizer Expand table <table><thead><tr><th>Property name</th><th>Type</th><th>Description</th></tr></thead><tbody><tr><td>DisplayName</td><td>String</td><td>The display name for the recognized intent; localized in the current language and to be shown in the DYM prompt (if applicable)</td></tr><tr><td>Score</td><td>Number</td><td>The recognizer score</td></tr><tr><td>TopicId</td><td>String</td><td>The Dataverse schema name of the topic</td></tr><tr><td>TriggerId</td><td>String</td><td>The unique ID of the trigger within the AdaptiveDialog topic</td></tr></tbody></table> For a complete list of Copilot Studio variables, including system variables, see Work with variables. https://learn.microsoft.com/en-us/microsoft-copilot-studio/advanced-clu-integration On information and belief, Microsoft continues to determine a context based on its score in Copilot	Property name	Type	Description	DisplayName	String	The display name for the recognized intent; localized in the current language and to be shown in the DYM prompt (if applicable)	Score	Number	The recognizer score	TopicId	String	The Dataverse schema name of the topic	TriggerId	String	The unique ID of the trigger within the AdaptiveDialog topic
Property name	Type	Description														
DisplayName	String	The display name for the recognized intent; localized in the current language and to be shown in the DYM prompt (if applicable)														
Score	Number	The recognizer score														
TopicId	String	The Dataverse schema name of the topic														
TriggerId	String	The unique ID of the trigger within the AdaptiveDialog topic														

Element	Analysis for Copilot
	as it did for Cortana. 5.4. Hypothesis Ranking and Selection HRS is the mechanism by which the platform ranks and selects the final response shown to the user. The processing is done in two stages, with an initial ranking of the hypotheses (HR) followed by selection of the best hypothesis according to rank score and other features (HS). One of the key decisions in dealing with open domain requests is the recognition of requests that are not handled by any specialized experience provider and letting such requests fall through to web search. AN OVERVIEW OF END-TO-END LANGUAGE UNDERSTANDING AND DIALOG MANAGEMENT FOR PERSONAL DIGITAL ASSISTANTS, Microsoft Corporation, 2016, available at https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/CortanaLUDialog-FromSLTproceedings.pdf (emphasis added) For example, Copilot uses the keyword “weather” to determine a weather-related context of a request.

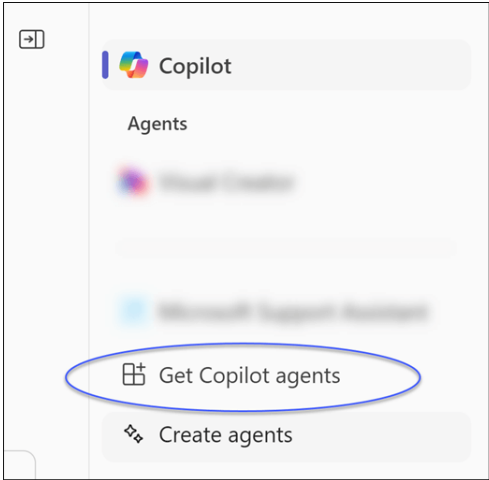
Element	Analysis for Copilot
	
[1.5] selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the	Copilot meets this element. Copilot's orchestrator selects a right skill based on the determined context.

Element	Analysis for Copilot
determined context;	Copilot orchestrator Between the user's natural language input and Copilot's natural language output, the Copilot orchestrator works behind the scenes to select and run the right skills from the right plugins for the user's task. The orchestration layer represents the interface between foundation Large Language Models (LLMs) and the many ways you can extend, enrich, and customize Copilot for the way your customers work.  The following diagram illustrates how the Microsoft 365 Copilot orchestrator selects the right plugin, with the right skill, at the right time, even when there are multiple options to choose from. 

Element	Analysis for Copilot
	3. Reasoning: The Copilot orchestrator formulates a plan comprised of multiple actions that it performs in an attempt to respond to the user's prompt. a. Context and tool selection: The orchestrator retrieves the user's conversation context from the context store and integrates data from Microsoft Graph to refine the context. It then adjusts the initial query based on this updated context and forwards it to the LLM (large language model) to guide the next steps. The LLM might proceed to generating a response using Copilot's built-in capabilities, or it might determine that additional data is necessary. If more information is needed, the orchestrator does a search for the plugins (tools) with the right skill for the task from the user's enabled plugins based on the descriptions of the plugins and their functions. b. Function matching and parameter determination: The orchestrator formulates a new prompt that incorporates the user's initial query, the updated context, and the selected plugins, and presents it to the LLM. The LLM evaluates the input and specifies the optimal plugin and function within that plugin to address the task. It then provides the orchestrator with the necessary function details and parameters required to gather the needed information.

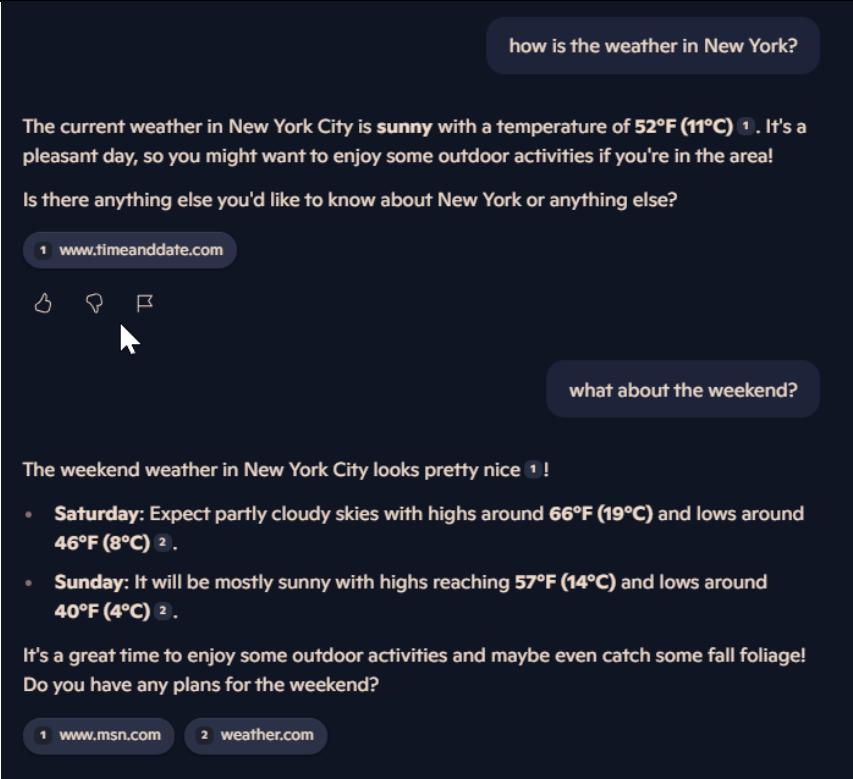
Element	Analysis for Copilot
	How Copilot's orchestrator matches plugins to user queries When a user submits a query to Copilot, the orchestrator searches its full catalog of skills (<i>functions</i>) from installed plugins to identify up to five skills that best match the query. The orchestrator first tries to match on exact words (lexical match) and expands its search scope as needed to include matches on descriptive meanings (semantic match), working from specific function names to general plugin descriptions, until all five function candidate slots are filled. Specifically, the following list shows the hierarchy of matching mechanisms for Copilot plugin function selection: 1. Lexical match on function name. 2. Semantic match on function description. 3. Lexical match on plugin name (adds all plugin functions to candidate list). 4. Semantic match on plugin name (adds all plugin functions to candidate list). The orchestrator works through this list until all five function candidate slots are filled. To best ensure that the Copilot orchestrator chooses your plugin to provide the right skill for users' queries, apply the Explore best practices for high-quality plugins. https://learn.microsoft.com/en-us/microsoft-365-copilot/extensibility/orchestrator (emphasis added) In addition, Copilot uses Copilot agents to process queries and tasks in a given domain or context. What are agents? In essence, agents are a scoped version of Copilot. They are designed to automate and execute specific business processes on your behalf. Each agent is an expert in one or two specific tasks, such as retrieving information, summarizing data, and even taking actions like sending emails or updating records. Agents can be customized to meet specific business needs, such as: • Human Resources agent • Analytics agent • Project management agent • Legal agent • Image creation agent

Element	Analysis for Copilot
	https://learn.microsoft.com/ko-kr/microsoft-365-copilot/extensibility/ New Copilot agents: Supercharge Microsoft 365 Copilot ClintWilliams  MICROSOFT Oct 31, 2024 Are you ready to supercharge Microsoft 365 Copilot? Following the recent Wave 2 announcements and the general availability of Copilot agents, we are excited to introduce a curated set of new Copilot agents developed by Microsoft. These agents are designed to help you and your team optimize and reinvent your business processes. https://techcommunity.microsoft.com/blog/microsoft365copilotblog/new-copilot-agents-supercharge-microsoft-365-copilot/4266995

Element	Analysis for Copilot
	Explore agents Agents can be found in the app store in Microsoft Teams and the Microsoft 365 app. From Copilot in Teams or the Microsoft 365 app, select Get Copilot agents on the side panel to the right to explore agents in the app store.  https://support.microsoft.com/en-us/topic/get-started-with-agents-for-microsoft-365-copilot-169469d7-328d-4d37-9090-bfc2058a39bd

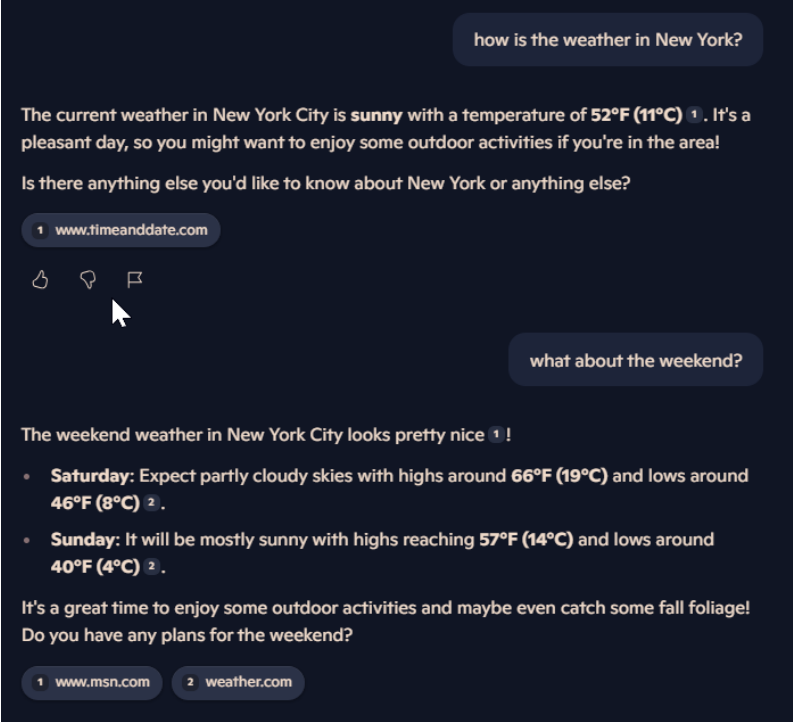
Element	Analysis for Copilot									
	How do you want to extend Microsoft Copilot? Extend in Microsoft Copilot directly Extend with declarative agents What are the features you want to build? <table><thead><tr><th>FEATURES</th><th>PRO CODE TOOLS</th><th>LOW CODE TOOLS</th></tr></thead><tbody><tr><td>KNOWLEDGE SharePoint files One Drive files Graph connectors</td><td>Teams Toolkit for VS Code</td><td>Copilot Studio Copilot Studio</td></tr><tr><td>ACTIONS *API plugins Prompts Flow plugins Power Platform connector plugins **Message Extensions</td><td>Teams Toolkit for VS Code</td><td>Copilot Studio Copilot Studio Copilot Studio Copilot Studio</td></tr></tbody></table> *API plugins only work with declarative agents ** Feature unavailable for declarative agents (Pro code and Low code tools works individually) https://learn.microsoft.com/sr-cyrl-rs/microsoft-365-copilot/extensibility/decision-guide What are declarative agents? Declarative agents are customized versions of Microsoft 365 Copilot that enable customers to create personalized experiences that run in Microsoft 365 Copilot. This feature builds on top of the Microsoft 365 Copilot system. Users can build declarative agents with capabilities and instructions on their purpose and behavior. Users can also add the option to access enterprise knowledge (including SharePoint and Graph Connectors). Read more about declarative agents: Declarative agents for Microsoft 365. Read more about overall Copilot Extensibility options: Extend Microsoft 365 Copilot. https://learn.microsoft.com/en-us/microsoft-365-copilot/extensibility/transparency-faq-declarative-agent; https://learn.microsoft.com/en-us/microsoft-365-copilot/extensibility/overview-declarative-agent#tailor-declarative-agents-for-your-scenario; https://learn.microsoft.com/en-us/microsoft-365-copilot/extensibility/overview-declarative-agent#tailor-declarative-agents-for-your-scenario	FEATURES	PRO CODE TOOLS	LOW CODE TOOLS	KNOWLEDGE SharePoint files One Drive files Graph connectors	Teams Toolkit for VS Code	Copilot Studio Copilot Studio	ACTIONS *API plugins Prompts Flow plugins Power Platform connector plugins **Message Extensions	Teams Toolkit for VS Code	Copilot Studio Copilot Studio Copilot Studio Copilot Studio
FEATURES	PRO CODE TOOLS	LOW CODE TOOLS								
KNOWLEDGE SharePoint files One Drive files Graph connectors	Teams Toolkit for VS Code	Copilot Studio Copilot Studio								
ACTIONS *API plugins Prompts Flow plugins Power Platform connector plugins **Message Extensions	Teams Toolkit for VS Code	Copilot Studio Copilot Studio Copilot Studio Copilot Studio								

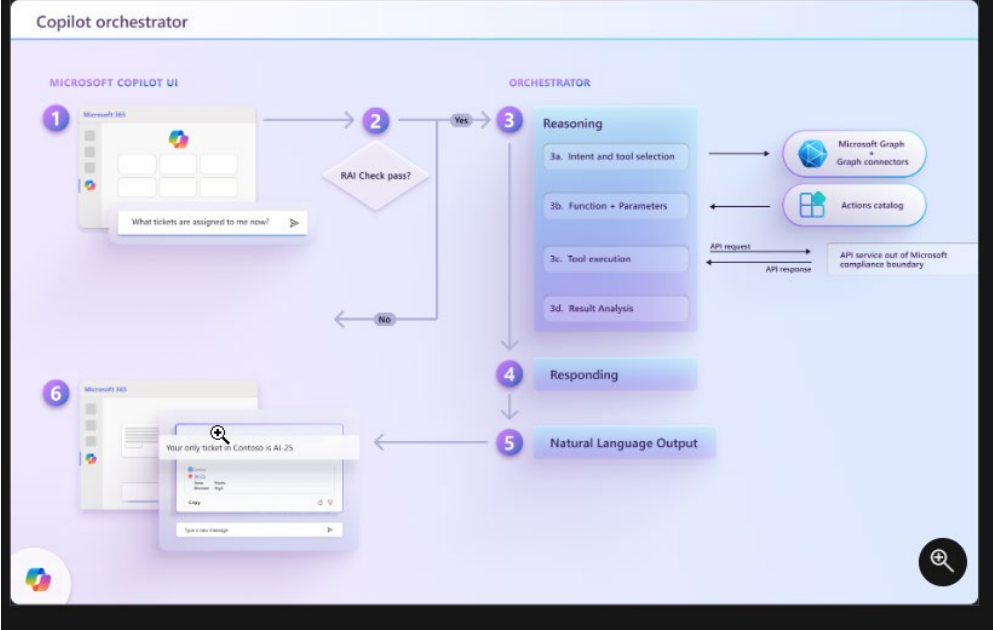
Element	Analysis for Copilot
	copilot/extensibility/agents-are-apps (emphasis added) On information and belief, Copilot can select Copilot agent based on the determined context. Copilot is your personal, private assistant that works solely for you, enhancing your capabilities. And agents are expert systems that operate autonomously on behalf of a process or company. You configure, secure, and test your agent, and then it automates for you and your team across apps and data sources. We expect every employee will have Copilot and will be supported by many agents. Where agents thrive in the AI world Organizations are always looking for ways to enhance efficiency, improve customer experience, and drive growth, but identifying the starting point is crucial. Copilot Studio can help you modernize existing processes or create new ones, paving the way for innovation and unlocking additional business value. Organizations like Pets at Home have created autonomous agents for specific business workflows in Copilot Studio. Pets at Home, the UK's leading pet care business, created an agent for its profit protection team to more efficiently compile cases for skilled human review, which could have the potential to drive a seven-figure annual savings. https://www.microsoft.com/en-us/microsoft-copilot/blog/copilot-studio/unlocking-autonomous-agent-capabilities-with-microsoft-copilot-studio/ For example, on information and belief, for the request “what about the weekend?”, Copilot selects a weather domain agent depending on the context.

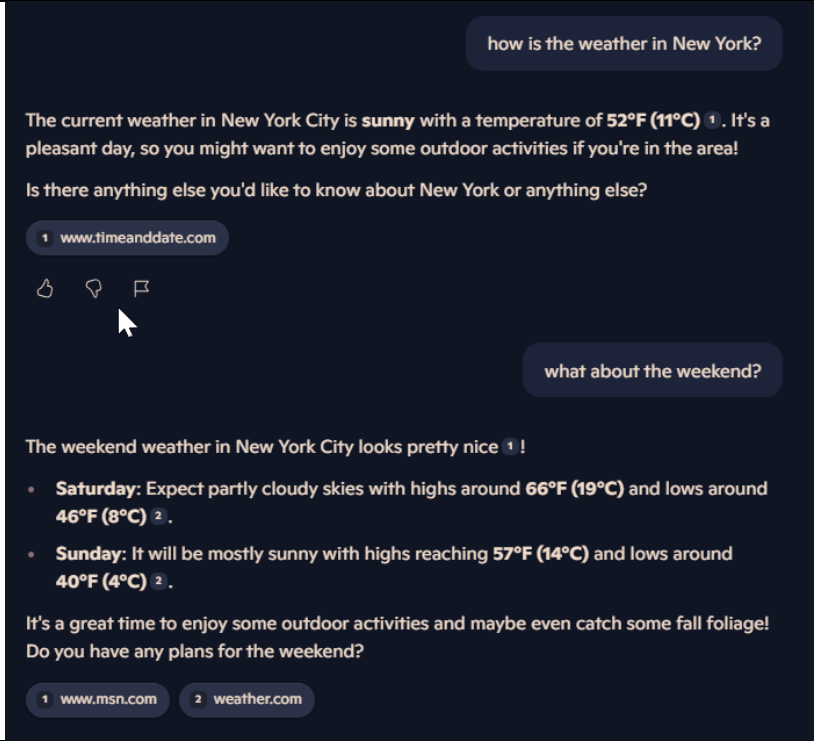
Element	Analysis for Copilot
	 how is the weather in New York? The current weather in New York City is sunny with a temperature of 52°F (11°C) ¹ . It's a pleasant day, so you might want to enjoy some outdoor activities if you're in the area! Is there anything else you'd like to know about New York or anything else? ¹ www.timeanddate.com what about the weekend? The weekend weather in New York City looks pretty nice ¹ ! • Saturday: Expect partly cloudy skies with highs around 66°F (19°C) and lows around 46°F (8°C) ² . • Sunday: It will be mostly sunny with highs reaching 57°F (14°C) and lows around 40°F (4°C) ² . It's a great time to enjoy some outdoor activities and maybe even catch some fall foliage! Do you have any plans for the weekend? ¹ www.msn.com ² weather.com
[1.6] formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context;	Copilot meets this element. For example, Copilot sends a query or command based on the natural language utterance in the form of an API request to the selected skill.

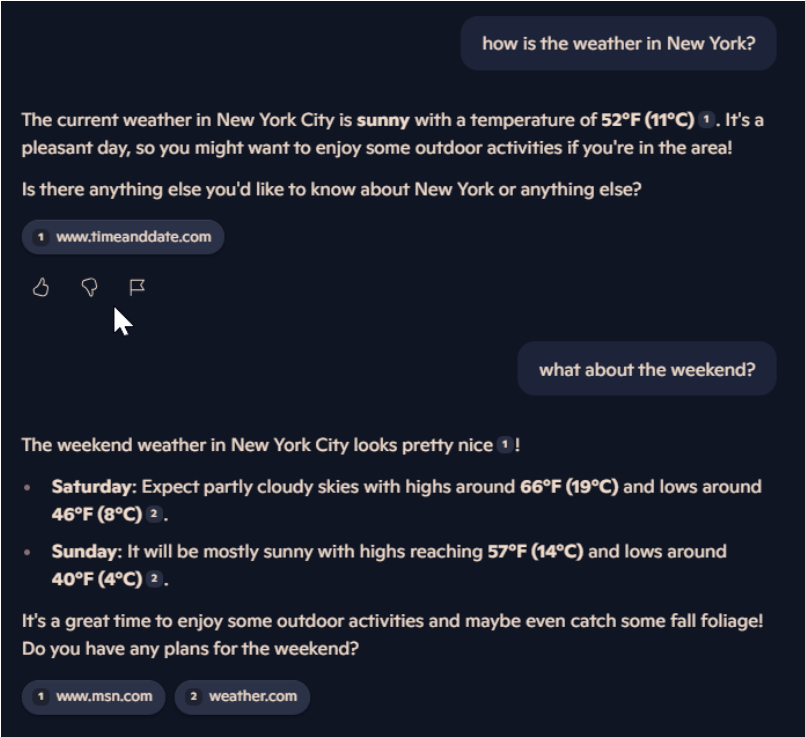
Element	Analysis for Copilot
	c. Tool initiation: The orchestrator uses the response from the LLM to construct an API request and send the request to the tool initiator, which securely retrieves the requested information located outside of Copilot's infrastructure. It runs the request and sends the results back to the orchestrator for further processing. https://learn.microsoft.com/en-us/microsoft-365-copilot/extensibility/orchestrator (emphasis added) For example, On information and belief, Copilot formulates a user request according to a grammar in a similar way as Cortana, and Cortana formulates a user request according to a grammar that uses slots and entities. 5.1. Slot/Entity Carry Over (SCO) and Co-reference Tracking keywords contained in slots and associated entities enables a certain set of user initiative experiences, <i>e.g.</i>

Element	Analysis for Copilot
	• Turn 1: “find french restaurants in seattle” • State 1: cuisine=“french”, place_type=“restaurants”, absolute_location=“seattle” • Turn 2: “how about chinese” • State 2: cuisine=“chinese”, place_type=“restaurants”, absolute_location=“seattle” SCO decides which slots from previous turns are still relevant in the current turn of a multi-turn conversation. It also models implicit start-overs, <i>i.e.</i> dropping of all slots, and operates both within and across domains powering interactions such as “What’s the weather like at Snow Lake?”, “Get me driving directions.” We use a hybrid rule-model approach to implement our SCO system, with the set of rules serving as baseline with a generalized ML model to scale better to new scenarios. Model-based SCO is implemented as a supervised learning task [22]. For each slot from previous turns, the model decides to carry-over or drop slots. Slots in the current turn are unconditionally kept. Our implementation of SCO is as a gradient boosted decision tree, where the concatenation of the current and previous turn utterances are compared to indicate relevant slots. At least one most likely tagging is passed on for further processing. If multiple taggings are produced by SCO, the hypothesis ranking and selection (HRS) component is the ultimate determiner of the winning tagging. For experiences that use strongly typed entities, <i>e.g.</i> celebrity question-answer experiences, an alternative co-reference resolution model exists that learns relationships such as ‘him’ or ‘her’ from mining knowledge bases, <i>e.g.</i> Satori – Microsoft’s equivalent of Freebase or Google’s Knowledge Graph. AN OVERVIEW OF END-TO-END LANGUAGE UNDERSTANDING AND DIALOG MANAGEMENT FOR PERSONAL DIGITAL ASSISTANTS, Microsoft Corporation, 2016, available at https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/CortanaLUDialog-FromSLTproceedings.pdf For example, on information and belief, Copilot formulates a weather-related request in accordance with a grammar used by the selected weather domain agent, which request may need to include, for example, location information and time information.

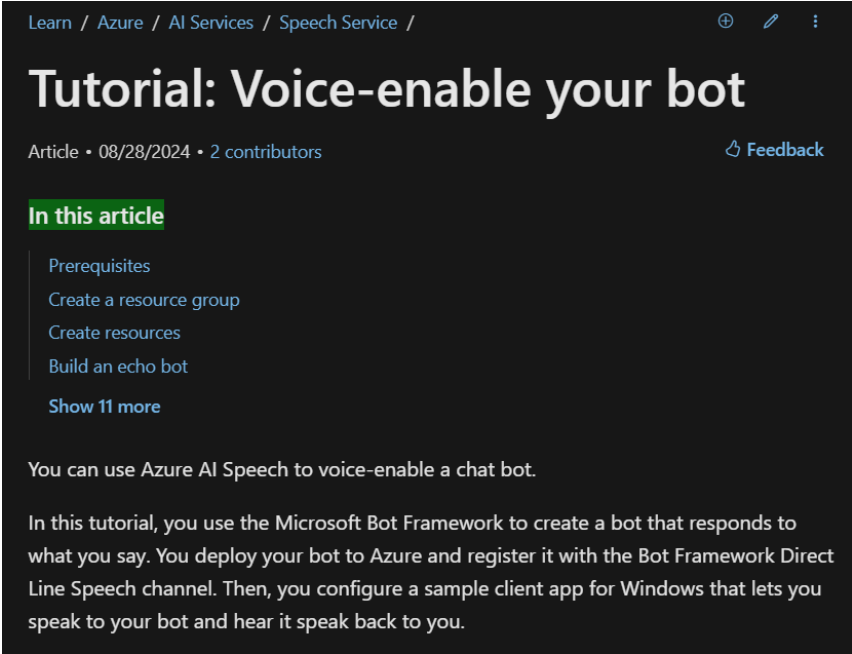
Element	Analysis for Copilot
	 how is the weather in New York? The current weather in New York City is sunny with a temperature of 52°F (11°C) ¹. It's a pleasant day, so you might want to enjoy some outdoor activities if you're in the area! Is there anything else you'd like to know about New York or anything else? ¹ www.timeanddate.com what about the weekend? The weekend weather in New York City looks pretty nice ¹! • Saturday: Expect partly cloudy skies with highs around 66°F (19°C) and lows around 46°F (8°C) ². • Sunday: It will be mostly sunny with highs reaching 57°F (14°C) and lows around 40°F (4°C) ². It's a great time to enjoy some outdoor activities and maybe even catch some fall foliage! Do you have any plans for the weekend? ¹ www.msn.com ² weather.com
[1.7] invoking the selected domain agent to process the formulated request; and	Copilot meets this element. For example, Copilot uses the orchestrator to invoke the selected skill to process the formulated request.

Element	Analysis for Copilot
	The following diagram illustrates how the Microsoft 365 Copilot orchestrator selects the right plugin, with the right skill, at the right time, even when there are multiple options to choose from.  The diagram, titled "Copilot orchestrator", illustrates the workflow of the Microsoft 365 Copilot. It is divided into two main sections: "MICROSOFT COPILOT UI" and "ORCHESTRATOR". Step 1: The user interacts with the "MICROSOFT COPILOT UI", asking a question like "What tickets are assigned to me now?". Step 2: The request is processed through an "RAI Check pass?" decision point. If it fails, it loops back to the UI. If it passes, it proceeds to the orchestrator. Step 3: Reasoning (within the ORCHESTRATOR): 3a. Intent and tool selection: Connects to "Microsoft Graph + Graph connectors". 3b. Function + Parameters: Connects to an "Actions catalog". 3c. Tool execution: Sends an "API request" to an "API service out of Microsoft compliance boundary" and receives an "API response". 3d. Result Analysis: Receives data from the API response. Step 4: Responding Step 5: Natural Language Output Step 6: The final output is presented back to the "MICROSOFT COPILOT UI" as a response, such as "Your only ticket in Contoso is AI-25". c. Tool initiation: The orchestrator uses the response from the LLM to construct an API request and send the request to the tool initiator, which securely retrieves the requested information located outside of Copilot's infrastructure. It runs the request and sends the results back to the orchestrator for further processing. https://learn.microsoft.com/en-us/microsoft-365-copilot/extensibility/orchestrator (emphasis added) For example, Copilot invokes the selected weather domain agent to process a formulated weather-related request.

Element	Analysis for Copilot
	 how is the weather in New York? The current weather in New York City is sunny with a temperature of 52°F (11°C) ¹. It's a pleasant day, so you might want to enjoy some outdoor activities if you're in the area! Is there anything else you'd like to know about New York or anything else? ¹ www.timeanddate.com what about the weekend? The weekend weather in New York City looks pretty nice ¹! • Saturday: Expect partly cloudy skies with highs around 66°F (19°C) and lows around 46°F (8°C) ². • Sunday: It will be mostly sunny with highs reaching 57°F (14°C) and lows around 40°F (4°C) ². It's a great time to enjoy some outdoor activities and maybe even catch some fall foliage! Do you have any plans for the weekend? ¹ www.msn.com ² weather.com
[1.8] presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.	Copilot meets this element. Copilot presents the results of the processed request in the form of natural language output. 4. Responding: The orchestrator compiles all the information gathered during the reasoning process and submits it to the LLM to create a final response. After ensuring the response complies with Responsible AI guidelines, it sends the response back to the orchestrator, which logs it in the context store and delivers it to the user via the Copilot UI. 5. Natural language output: Finally, the orchestrator delivers the response to the user and updates the conversation state. Copilot is ready for its next prompt.

Element	Analysis for Copilot
	https://learn.microsoft.com/en-us/microsoft-365-copilot/extensibility/orchestrator For example, Cortana presents the results of the processed weather-related request to the user.  The screenshot displays a chat window with a dark background. At the top, a user input bubble contains the text "how is the weather in New York?". Below this, the system response provides current weather information for New York City, stating it is sunny with a temperature of 52°F (11°C). It includes a citation marker "1" and a follow-up question: "Is there anything else you'd like to know about New York or anything else?". A citation bubble below the response shows "1 www.timeanddate.com". Below the citation are three icons: a thumbs up, a speech bubble, and a flag. A second user input bubble at the bottom asks "what about the weekend?". The system response provides a weekend forecast for New York City, mentioning "pretty nice" weather and specific forecasts for Saturday (partly cloudy, 66°F/19°C high, 46°F/8°C low) and Sunday (mostly sunny, 57°F/14°C high, 40°F/4°C low). It includes citation markers "1" and "2". A final citation bubble at the bottom shows "1 www.msn.com" and "2 weather.com".

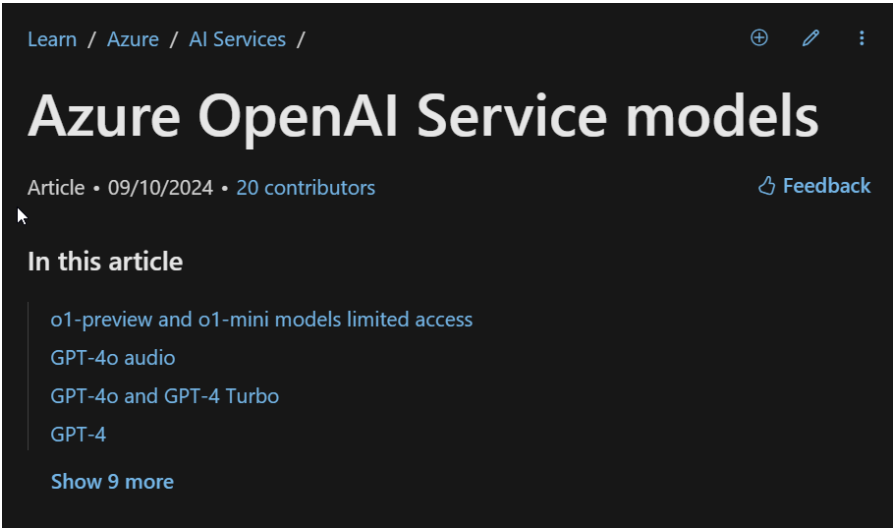
Azure AI and Azure OpenAI

Element	Analysis for Azure AI and Azure OpenAI
[1.0]. A method responsive to a user generated natural language speech utterance, comprising:	Azure AI and Azure OpenAI meet this element.  https://learn.microsoft.com/en-us/azure/ai-services/speech-service/tutorial-voice-enable-your-bot-speech-sdk On information and belief, Azure AI uses Language Understanding (LUIS) and Conversational Language Understanding technology to understand and respond to user speech utterances.

Element	Analysis for Azure AI and Azure OpenAI
	Combine LUIS and question answering capabilities Article • 08/28/2024 • 1 contributor  Feedback In this article  - When to use each feature - Create an orchestration project - Set up orchestration between Azure AI services features - Change a LUIS resource to a language resource: - Next steps https://learn.microsoft.com/en-us/azure/ai-services/luis/how-to/orchestration-projects

Element	Analysis for Azure AI and Azure OpenAI
	What is conversational language understanding? Article • 08/28/2024 • 4 contributors Feedback In this article - Example usage scenarios - Project development lifecycle - Reference documentation and code samples - Responsible AI - Next steps Conversational language understanding is one of the custom features offered by Azure AI Language. It is a cloud-based API service that applies machine-learning intelligence to enable you to build natural language understanding component to be used in an end-to-end conversational application. https://learn.microsoft.com/en-us/microsoft-copilot-studio/advanced-clu-integration
[1.1] receiving the user generated natural language speech utterance, the received user utterance containing at least one request;	Azure AI and Azure OpenAI meet this element. See element [1.1].
[1.2] maintaining a dynamic set of prior probabilities or fuzzy	Azure AI and Azure OpenAI meet this element. For example, Azure Cognitive Search supports fuzzy search.

Element	Analysis for Azure AI and Azure OpenAI
possibilities usable at each stage of processing the received user utterance;	Azure Cognitive Search supports fuzzy search, a type of query that compensates for typos and misspelled terms in the input string. It does this by scanning for terms having a similar composition. Expanding search to cover near-matches has the effect of auto-correcting a typo when the discrepancy is just a few misplaced characters. https://data-driven.com/blog/fuzzy-search-using-azure-cognitive-search/ For example, Azure AI uses machine learning and AI models (e.g., GPT), which on information and belief use prior probabilities or fuzzy possibilities to understand user intent and references. What is key phrase extraction in Azure AI Language? Article • 09/27/2024 • 4 contributors Feedback In this article - Typical workflow - Get started with Key phrase extraction - Reference documentation and code samples - Responsible AI - Next steps Key phrase extraction is one of the features offered by Azure AI Language, a collection of machine learning and AI algorithms in the cloud for developing intelligent applications that involve written language. Use key phrase extraction to quickly identify the main concepts in text. For example, in the text <i>"The food was delicious and the staff were wonderful."</i>, key phrase extraction will return the main topics: <i>"food"</i> and <i>"wonderful staff"</i>.

Element	Analysis for Azure AI and Azure OpenAI
	https://learn.microsoft.com/en-us/azure/ai-services/language-service/keyphrase-extraction/overview  https://learn.microsoft.com/en-gb/azure/ai-services/openai/concepts/models?tabs=python-secure
[1.3]. recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;	Azure AI and Azure OpenAI meet this element. For example, Azure AI uses speech recognition technology (e.g., Universal Language Model) and natural language processing (NLP) technology to recognize words in user utterances.

Element	Analysis for Azure AI and Azure OpenAI
	Out of the box, speech recognition utilizes a Universal Language Model as a base model that is trained with Microsoft-owned data and reflects commonly used spoken language. The base model is pretrained with dialects and phonetics representing various common domains. When you make a speech recognition request, the most recent base model for each supported language is used by default. The base model works well in most speech recognition scenarios. Custom speech allows you to tailor the speech recognition model to better suit your application's specific needs. This can be particularly useful for: • Improving recognition of domain-specific vocabulary: Train the model with text data relevant to your field. • Enhancing accuracy for specific audio conditions: Use audio data with reference transcriptions to refine the model. https://learn.microsoft.com/en-au/azure/ai-services/speech-service/speech-to-text What is Azure AI Language? Article • 08/28/2024 • 5 contributors Feedback In this article - Available features - Which Language service feature should I use? - Migrate from Text Analytics, QnA Maker, or Language Understanding (LUIS) - Tutorials Show 3 more Azure AI Language is a cloud-based service that provides Natural Language Processing (NLP) features for understanding and analyzing text. Use this service to help build intelligent applications using the web-based Language Studio, REST APIs, and client libraries. https://learn.microsoft.com/en-us/azure/ai-services/language-service/overview?tabs=sentiment

Element	Analysis for Azure AI and Azure OpenAI
	On information and belief, Azure AI recognizes words and phrases using information relating to a user. What data does Azure AI Language process and how does it process it? • Azure AI Language processes text data that is sent by the customer to the system for the purposes of getting a response from one of the available features. • All results of the requested feature are sent back to the customer in the API response as specified in the API reference. For example, if Language Detection is requested, the language code is returned along with a confidence score for each text record. • Azure AI Language uses aggregate telemetry such as which APIs are used and the number of calls from each subscription and resource for service monitoring purposes. • Azure AI Language doesn't store or process customer data outside the region where the customer deploys the service instance. • Azure AI Language encrypts all content, including customer data, at rest. https://learn.microsoft.com/cs-cz/legal/cognitive-services/language-service/data-privacy

Element	Analysis for Azure AI and Azure OpenAI
	How does the Azure OpenAI Service process data? The diagram below illustrates how your data is processed. This diagram covers several types of processing: 1. How the Azure OpenAI Service processes your prompts via inferencing to generate content (including when additional data from a designated data source is added to a prompt using Azure OpenAI on your data, Assistants, or batch processing). 2. How the Assistants feature stores data in connection with Messages, Threads, and Runs. 3. How the Batch feature processes your uploaded data. 4. How the Azure OpenAI Service creates a fine-tuned (custom) model with your uploaded data. 5. How the Azure OpenAI Service and Microsoft personnel analyze prompts and completions (text and image) for harmful content and for patterns suggesting the use of the service in a manner that violates the Code of Conduct or other applicable product terms. https://learn.microsoft.com/en-us/legal/cognitive-services/openai/data-privacy?WT.mc_id=academic-97358-cacaste&context=%2Fazure%2Fcognitive-services%2Fopenai%2Fcontext%2Fcontext&tabs=azure-portal
[1.4] parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases;	Azure AI and Azure OpenAI meet this element. For example, Azure AI uses key phrase extraction to parse user input.

Element	Analysis for Azure AI and Azure OpenAI
	What is key phrase extraction in Azure AI Language? Article • 09/27/2024 • 4 contributors Feedback In this article - Typical workflow - Get started with Key phrase extraction - Reference documentation and code samples - Responsible AI - Next steps Key phrase extraction is one of the features offered by Azure AI Language, a collection of machine learning and AI algorithms in the cloud for developing intelligent applications that involve written language. Use key phrase extraction to quickly identify the main concepts in text. For example, in the text "<i>The food was delicious and the staff were wonderful.</i>", key phrase extraction will return the main topics: "<i>food</i>" and "<i>wonderful staff</i>". This documentation contains the following types of articles: Quickstarts are getting-started instructions to guide you through making requests to the service. How-to guides contain instructions for using the service in more specific or customized ways. https://learn.microsoft.com/en-us/azure/ai-services/language-service/key-phrase-extraction/overview#reference-documentation-and-code-samples For example, Azure AI recognizes intents for user utterances.

Element	Analysis for Azure AI and Azure OpenAI
	Conversational language understanding (CLU) is a cloud-based AI offering in Azure AI Language. It's the newest generation of Language Understanding (LUIS) and offers backwards compatibility with previously created LUIS applications. CLU employs state-of-the-art machine learning intelligence to allow users to build a custom natural language understanding model for predicting intents and entities in conversational utterances. CLU offers the following advantages over LUIS: • Improved accuracy with state-of-the-art machine learning models for better intent classification and entity extraction. LUIS required more examples to generalize certain concepts in intents and entities, while CLU's more advanced machine learning reduces the burden on customers by requiring significantly less data. https://learn.microsoft.com/en-us/azure/ai-services/language-service/conversational-language-understanding/how-to/migrate-from-luis?context=%2Fazure%2Fai-services%2FLUIS%2Fcontext%2Fcontext&tabs=luis-portal (emphasis added) Azure AI also recognizes and tracks contexts for user utterances.

Element	Analysis for Azure AI and Azure OpenAI
	DialogContext class Reference  Feedback Package: botbuilder-dialogs The context for the current dialog turn with respect to a specific DialogSet. Remarks This includes the turn context, information about the dialog set, and the state of the dialog stack. From code outside of a dialog in the set, use DialogSet.createContext to create the dialog context. Then use the methods of the dialog context to manage the progression of dialogs in the set. When you implement a dialog, the dialog context is a parameter available to the various methods you override or implement. https://learn.microsoft.com/en-us/javascript/api/botbuilder-dialogs/dialogcontext?view=botbuilder-ts-latest

Element	Analysis for Azure AI and Azure OpenAI
	Dialog state Dialogs can implement a <i>multi-turn conversation</i>, and as such, they rely on <i>persisted state</i> across turns. Without state in dialogs, your bot wouldn't know where it was in the conversation or what information it had already gathered. To retain a dialog's place in the conversation, the dialog's state must be retrieved from and saved to memory each turn. This is handled via a dialog state property accessor defined on the bot's conversation state. This dialog state manages information for all active dialogs, and children of active dialogs. This allows the bot to pick up where it left off last and to handle various conversation models. At run time, the dialog state property includes information on where the dialog is in its logical process, including any internally collected information in the form of a <i>dialog instance</i> object. Again, this needs to be read into the bot and saved out to memory each turn.

Element	Analysis for Azure AI and Azure OpenAI										
	Dialog infrastructure Along with various types of dialogs, the following classes are involved in the design and control of conversations. While you don't usually need to interact with these classes directly, being aware of them and their purpose is useful when designing dialogs for a bot.  Expand table <table> <tr> <th>Class</th><th>Description</th></tr> <tr> <td><i>Dialog set</i></td><td>Defines a collection of dialogs that can reference each other and work in concert.</td></tr> <tr> <td><i>Dialog context</i></td><td>Contains information about all active dialogs.</td></tr> <tr> <td><i>Dialog instance</i></td><td>Contains information about one active dialog.</td></tr> <tr> <td><i>Dialog turn result</i></td><td>Contains status information from an active, or recently active, dialog. If the active dialog ended, this contains its return value.</td></tr> </table>	Class	Description	<i>Dialog set</i>	Defines a collection of dialogs that can reference each other and work in concert.	<i>Dialog context</i>	Contains information about all active dialogs.	<i>Dialog instance</i>	Contains information about one active dialog.	<i>Dialog turn result</i>	Contains status information from an active, or recently active, dialog. If the active dialog ended, this contains its return value.
Class	Description										
<i>Dialog set</i>	Defines a collection of dialogs that can reference each other and work in concert.										
<i>Dialog context</i>	Contains information about all active dialogs.										
<i>Dialog instance</i>	Contains information about one active dialog.										
<i>Dialog turn result</i>	Contains status information from an active, or recently active, dialog. If the active dialog ended, this contains its return value.										

Element	Analysis for Azure AI and Azure OpenAI
	The dialog stack A dialog context contains information about all active dialogs and includes a <i>dialog stack</i>, which acts as a <i>call stack</i> for all the active dialogs. Each container dialog has an inner set of dialogs that it's controlling, and so each active container dialog introduces an inner dialog context and dialog stack as part of its state. While you won't access the stack directly, understanding that it exists and its function will help you understand how various aspects of the dialogs library work. <a data-bbox="539 732 1908 764" href="https://learn.microsoft.com/en-us/azure/bot-service/bot-builder-concept-dialog?view=azure-bot-service-4.0">https://learn.microsoft.com/en-us/azure/bot-service/bot-builder-concept-dialog?view=azure-bot-service-4.0 Retain conversation context when interacting with an Azure OpenAI ChatGPT model  Subscribe  Like(6)  Share  Report Posted on 14 Mar 2023 10:44:55 by <a data-bbox="827 1073 919 1094" href="#">HenryJammes <a data-bbox="539 1101 1810 1166" href="https://community.powerplatform.com/forums/thread/details/?threadid=e4d204b9-32e1-4e72-9fbe-616e7741d1d5">https://community.powerplatform.com/forums/thread/details/?threadid=e4d204b9-32e1-4e72-9fbe-616e7741d1d5 Azure AI assigns confidence scores to potential intents.

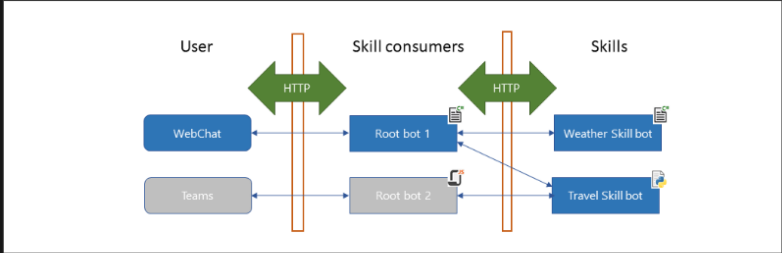
Element	Analysis for Azure AI and Azure OpenAI		
	Learn / Bot Builder TypeScript SDK / botbuilder-ai / IntentData interface Reference Package: botbuilder-ai Strongly typed information corresponding to a LUIS intent. Properties Expand table <table><tr><td>score</td><td>Confidence in intent classification on a scale from 0.0 - 1.0.</td></tr></table> https://learn.microsoft.com/en-us/javascript/api/botbuilder-ai/intentdata?view=botbuilder-ts-latest IntentScore interface Reference Package: botbuilder-core Score plus any extra information about an intent. https://learn.microsoft.com/en-us/javascript/api/botbuilder-core/intentscore?view=botbuilder-ts-latest	score	Confidence in intent classification on a scale from 0.0 - 1.0.
score	Confidence in intent classification on a scale from 0.0 - 1.0.		

Element	Analysis for Azure AI and Azure OpenAI						
	Return all intents' scores You assign an utterance to a single intent. When LUIS receives an utterance, by default it returns the top intent for that utterance. If you want the scores for all intents for the utterance, you can provide a flag in the query string of the prediction API. Expand table <table><thead><tr><th>Prediction API version</th><th>Flag</th></tr></thead><tbody><tr><td>V2</td><td><code>verbose=true</code></td></tr><tr><td>V3</td><td><code>show-all-intents=true</code></td></tr></tbody></table> https://learn.microsoft.com/en-us/azure/ai-services/LUIS/concepts/intents For example, for the utterance “what is the hottest month on new york,” Azure AI returns scores for possible intents.	Prediction API version	Flag	V2	<code>verbose=true</code>	V3	<code>show-all-intents=true</code>
Prediction API version	Flag						
V2	<code>verbose=true</code>						
V3	<code>show-all-intents=true</code>						

Element	Analysis for Azure AI and Azure OpenAI
	13. The prediction service returns data in JSON format. This includes a prediction of which intent your query most likely represents and an extraction of the entities that were in the query. Your application can use this data to understand what the user is trying to do <pre data-bbox="577 354 1134 1250"> { "query": "what is the hottest month on new york?", "prediction": { "topIntent": "Weather.CheckWeatherTime", "intents": { "Weather.CheckWeatherTime": { "score": 0.9982601 }, "Weather.GetWeatherAdvisory": { "score": 0.009934696 }, "Weather.QueryWeather": { "score": 0.009064876 }, "Weather.ChangeTemperatureUnit": { "score": 0.00195273571 }, "Weather.CheckWeatherValue": { "score": 0.000958713063 }, "None": { "score": 0.000537212065 } }, "entities": { "Weather.WeatherRange": ["hottest"], "Weather.Historic": ["month"], "geographyV2": [{ "value": "new york", "type": "city" }] } } } </pre> (JSON returned by prediction service) https://microsoft.github.io/AzureTipsAndTricks/blog/tip318.html

Element	Analysis for Azure AI and Azure OpenAI
[1.5] selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context;	Azure AI and Azure OpenAI meet this element. For example, Azure AI uses bots to process user queries. Create a bot After publishing, you can create a bot from the Publish page: - You can create several bots quickly, all pointing to the same knowledge base for different regions or pricing plans for the individual bots. - If you want only one bot for the knowledge base, use the View all your bots on the Azure portal link to view a list of your current bots. https://learn.microsoft.com/en-us/azure/ai-services/qnamaker/quickstarts/create-publish-knowledge-base <pre> graph TD subgraph Device_Channel [Device/Channel] Text Audio Video end subgraph Bot [Conversational app/bot] Input_recognition[Input recognition] Reasoning Response_generation[Response generation] end Other_services[Other services] User((User)) Device_Channel --> Input_recognition Input_recognition --> Reasoning Reasoning --> Response_generation Response_generation --> Device_Channel Bot <--> Other_services User --- Device_Channel </pre> https://learn.microsoft.com/en-us/azure/bot-service/bot-service-overview?view=azure-bot-service-4.0

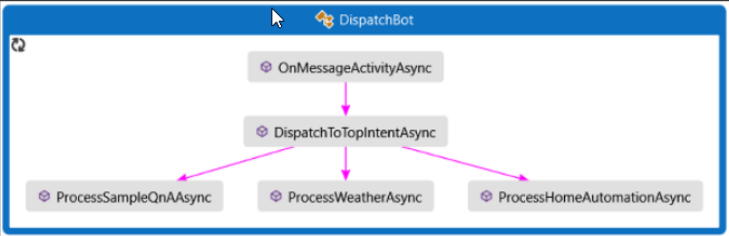
Element	Analysis for Azure AI and Azure OpenAI
	Build a robust B2C solution tailored to your specific operations In order to help our customers and partners understand and explore what's possible with Azure Communication Services and Azure AI, the Azure Communication Services team recorded an episode with MS Mechanics that takes you through a 'hero' scenario for customer intake. We also have all the code from the video available on Github for you to try. Imagine this scenario: you want to install solar panels on your roof, but you want to know more about rebates and offers that are available. You start a chat conversation on a website with a bot who tells you all the latest information, and even allows you to move the conversation from your browser to a live voice conversation on the phone. When you need more detailed assistance, that only a human technician can provide, your chat conversation is seamlessly handed-off to a live agent who has all the AI-assisted context and information that they need to help you. Finally, the agent uses AI to generate follow up emails and SMS summaries. This solution is powered by Azure Communication Services and Azure AI Service, and shows how to: 1. Connect Azure Communication Services Chat to Azure Cognitive Search using Azure OpenAI and ChatGPT to build a bot that can answer questions specific to a product domain. The solution uses some of the techniques described in ChatGPT + Enterprise data with Azure OpenAI and Cognitive Search such as Retrieval Augmented Generation (RAG) to create a ChatGPT-like experience using your own data knowledge base. https://techcommunity.microsoft.com/t5/azure-communication-services/accelerate-customer-outcomes-with-azure-ai-services-and-azure/ba-p/3937262 (emphasis added) Azure AI also uses skills to handle user queries. A skill is a bot that handles tasks in a specific domain.

Element	Analysis for Azure AI and Azure OpenAI
	Skills overview Article • 11/29/2022 • 6 contributors Feedback In this article Conceptual architecture Skill manifests Bot-to-bot communication Debugging skill conversations Additional information APPLIES TO: SDK v4 You can extend a bot using a <i>skill</i> bot. A skill can be consumed by various other bots, facilitating reuse, and in this way, you can create a user-facing bot and extend it by consuming your own or third-party skills. • A <i>skill</i> is a bot that can perform a set of tasks for another bot—a bot can be both a skill and a user-facing bot. • A <i>skill consumer</i> is a bot that can call one or more skills. A user-facing skill consumer is also called a <i>root bot</i>. • A <i>skill manifest</i> is a JSON file that describes the actions the skill can perform, its input and output parameters, and the skill's endpoints. ◦ Developers who don't have access to the skill's source code can use the information in the manifest to design their skill consumer. ◦ The <i>skill manifest schema</i> is a JSON file that describes the schema of the skill manifest. ◦ See how to implement a skill and how to write a skill manifest for sample skill manifests. This diagram shows some of the possible permutations.  <pre> graph LR subgraph User WebChat[WebChat] Teams[Teams] end subgraph Skill_consumers Root_bot_1[Root bot 1] Root_bot_2[Root bot 2] end subgraph Skills Weather_Skill_bot[Weather Skill bot] Travel_Skill_bot[Travel Skill bot] end WebChat <--> HTTP Root_bot_1 Teams <--> HTTP Root_bot_2 Root_bot_1 <--> HTTP Weather_Skill_bot Root_bot_1 <--> HTTP Travel_Skill_bot Root_bot_2 <--> HTTP Weather_Skill_bot Root_bot_2 <--> HTTP Travel_Skill_bot </pre>

Element	Analysis for Azure AI and Azure OpenAI
	https://learn.microsoft.com/en-us/azure/bot-service/skills-conceptual?view=azure-bot-service-4.0 Azure / Implement a skill Article • 10/16/2024 • 2 contributors Feedback In this article - Prerequisites - About this sample - Resources - Application configuration - Show 7 more APPLIES TO: SDK v4 You can use skills to extend another bot. A <i>skill</i> is a bot that can perform a set of tasks for another bot. • A manifest describes a skill's interface. Developers who don't have access to the skill's source code can use the information in the manifest to design their skill consumer. • A skill can use claims validation to manage which bots or users can access it. This article demonstrates how to implement a skill that echoes the user's input. https://docs.azure.cn/en-us/bot-service/skill-implement-skill?view=azure-bot-service-4.0&tabs=cs For example, on information and belief, Azure AI orchestration relates to selecting the right skill or bot.

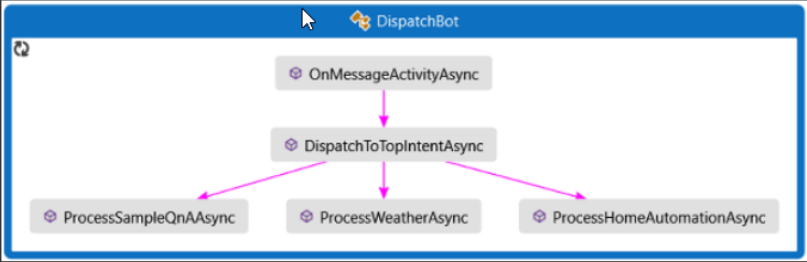
Element	Analysis for Azure AI and Azure OpenAI
	What is orchestration workflow? Article • 10/21/2024 • 5 contributors Feedback In this article Example usage scenarios Project development lifecycle Reference documentation and code samples Responsible AI Next steps Orchestration workflow is one of the features offered by Azure AI Language. It is a cloud-based API service that applies machine-learning intelligence to enable you to build orchestration models to connect Conversational Language Understanding (CLU), Question Answering projects and LUIS applications. By creating an orchestration workflow, developers can iteratively tag utterances, train and evaluate model performance before making it available for consumption. To simplify building and customizing your model, the service offers a custom web portal that can be accessed through the Language studio. You can easily get started with the service by following the steps in this quickstart. Example usage scenarios Orchestration workflow can be used in multiple scenarios across a variety of industries. Some examples are: Enterprise chat bot In a large corporation, an enterprise chat bot may handle a variety of employee affairs. It may be able to handle frequently asked questions served by a custom question answering knowledge base, a calendar specific skill served by conversational language understanding, and an interview feedback skill served by LUIS. The bot needs to be able to appropriately route incoming requests to the correct service. Orchestration workflow allows you to connect those skills to one project that handles the routing of incoming requests appropriately to power the enterprise bot. https://learn.microsoft.com/en-us/azure/ai-services/language-service/orchestration-workflow/overview

Element	Analysis for Azure AI and Azure OpenAI
	Bot Framework Orchestrator is an intent-only recognition engine. The Bot Framework CLI includes tools to generate a language model for Orchestrator from a collection of QnA Maker knowledge bases and LUIS language models. Your bot can then use Orchestrator to determine which service can best respond to the user's input. The Bot Framework SDK provides built-in support for LUIS and QnA Maker. This enables you to trigger dialogs or automatically answer questions using LUIS and QnA Maker with minimal configuration. https://learn.microsoft.com/en-us/azure/bot-service/bot-builder-tutorial-orchestrator?view=azure-bot-service-4.0&tabs=cs

Element	Analysis for Azure AI and Azure OpenAI
	About this sample This sample is based on a predefined set of LUIS and QnA Maker projects. However, to use QnA Maker in your bot, you need an existing knowledge base in the QnA Maker portal. Your bot then can use the knowledge base to answer the user's questions. For new bot development, consider using Copilot Studio. If you need to create a new knowledge base for a Bot Framework SDK bot, see the following Azure AI services articles: • What is question answering? • Create an FAQ bot • Azure Cognitive Language Services Question Answering client library for .NET C# JavaScript  <pre> graph TD OnMessageActivityAsync[OnMessageActivityAsync] --> DispatchToTopIntentAsync[DispatchToTopIntentAsync] DispatchToTopIntentAsync --> ProcessSampleQnAAsync[ProcessSampleQnAAsync] DispatchToTopIntentAsync --> ProcessWeatherAsync[ProcessWeatherAsync] DispatchToTopIntentAsync --> ProcessHomeAutomationAsync[ProcessHomeAutomationAsync] </pre> <code>OnMessageActivityAsync</code> is called for each user input received. This module finds the top scoring user intent and passes that result on to <code>DispatchToTopIntentAsync</code>. <code>DispatchToTopIntentAsync</code>, in turn, calls the appropriate app handler. • <code>ProcessSampleQnAAsync</code> - for bot FAQ questions. • <code>ProcessWeatherAsync</code> - for weather queries. • <code>ProcessHomeAutomationAsync</code> - for home lighting commands. The handler calls the LUIS or QnA Maker service and returns the generated result back to the user.

Element	Analysis for Azure AI and Azure OpenAI
	Call the services from your bot For each input from your user, the bot logic passes in user input to Orchestrator Recognizer, finds the top returned intent, and uses that information to call the appropriate service for the input. C# JavaScript In the <code>DispatchBot.cs</code> file whenever the <code>OnMessageActivityAsync</code> method is called, we check the incoming user message and get the top intent from Orchestrator Recognizer. We then pass the <code>topIntent</code> and <code>recognizerResult</code> on to the correct method to call the service and return the result. <code>bots\DispatchBot.cs</code> C# (archived) https://learn.microsoft.com/en-us/azure/bot-service/bot-builder-tutorial-orchestrator?view=azure-bot-service-4.0&tabs=cs
[1.6] formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context;	Azure AI and Azure OpenAI meet this element. On information and belief, Azure AI can formulate a request according to a grammar used by a bot or skill. Use multiple LUIS and QnA models with Orchestrator

Element	Analysis for Azure AI and Azure OpenAI
	If a bot uses multiple Language Understanding (LUIS) models and QnA Maker knowledge bases, you can use Bot Framework Orchestrator to determine which LUIS model or QnA Maker knowledge base best matches the user input. You can use the <code>bf orchestrator</code> CLI command to create an Orchestrator snapshot file, then use the snapshot file to route user input to the correct model at run time. This article describes how to use an <i>existing</i> QnA Maker knowledge base with Orchestrator. • For new bots, consider using the question answering and orchestration workflow features of Azure AI Language. • For more information about Orchestrator, see Intent recognition with Orchestrator in Composer. • For more information about the <code>bf orchestrator</code> command, see the Bot Framework CLI README. https://learn.microsoft.com/en-us/azure/bot-service/bot-builder-tutorial-orchestrator?view=azure-bot-service-4.0&tabs=cs

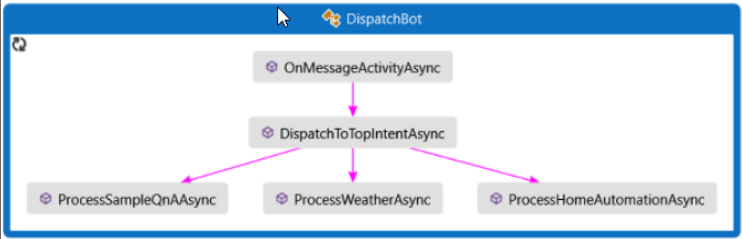
Element	Analysis for Azure AI and Azure OpenAI
	About this sample This sample is based on a predefined set of LUIS and QnA Maker projects. However, to use QnA Maker in your bot, you need an existing knowledge base in the QnA Maker portal. Your bot then can use the knowledge base to answer the user's questions. For new bot development, consider using Copilot Studio. If you need to create a new knowledge base for a Bot Framework SDK bot, see the following Azure AI services articles: • What is question answering? • Create an FAQ bot • Azure Cognitive Language Services Question Answering client library for .NET C# JavaScript  <pre> graph TD OnMessageActivityAsync[OnMessageActivityAsync] --> DispatchToTopIntentAsync[DispatchToTopIntentAsync] DispatchToTopIntentAsync --> ProcessSampleQnAAsync[ProcessSampleQnAAsync] DispatchToTopIntentAsync --> ProcessWeatherAsync[ProcessWeatherAsync] DispatchToTopIntentAsync --> ProcessHomeAutomationAsync[ProcessHomeAutomationAsync] </pre> <code>OnMessageActivityAsync</code> is called for each user input received. This module finds the top scoring user intent and passes that result on to <code>DispatchToTopIntentAsync</code>. <code>DispatchToTopIntentAsync</code>, in turn, calls the appropriate app handler. • <code>ProcessSampleQnAAsync</code> - for bot FAQ questions. • <code>ProcessWeatherAsync</code> - for weather queries. • <code>ProcessHomeAutomationAsync</code> - for home lighting commands. The handler calls the LUIS or QnA Maker service and returns the generated result back to the user.

Element	Analysis for Azure AI and Azure OpenAI
	Call the services from your bot For each input from your user, the bot logic passes in user input to Orchestrator Recognizer, finds the top returned intent, and uses that information to call the appropriate service for the input. C# JavaScript In the <code>DispatchBot.cs</code> file whenever the <code>OnMessageActivityAsync</code> method is called, we check the incoming user message and get the top intent from Orchestrator Recognizer. We then pass the <code>topIntent</code> and <code>recognizerResult</code> on to the correct method to call the service and return the result. <code>bots\DispatchBot.cs</code> C# (archived) https://learn.microsoft.com/en-us/azure/bot-service/bot-builder-tutorial-orchestrator?view=azure-bot-service-4.0&tabs=cs For example, Azure AI formulates a weather-related request according to the grammar used by the Weather API. Step 3: Integrating the Weather API 1. Add Weather API Integration: 1. In your bot's main dialog (e.g., <code>WeatherDialog.cs</code>), create a method to call the weather API. 2. Integrate CLU 1. Modify the JSON file that defines intents, entities, and utterances of the weather. 3. Implement the Weather API Call: 1. Use <code>RestService</code> library to send a request to the weather API and parse the response. 4. Handle Weather Response: 1. Create a modal class <code>WeatherData.cs</code> to parse the resulting API into the object.

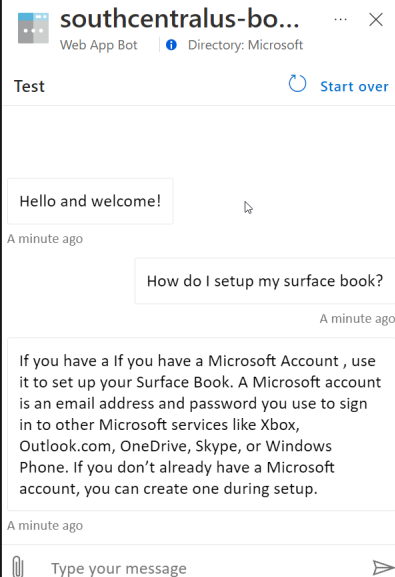
Element	Analysis for Azure AI and Azure OpenAI
	• Implement Weather API Call <pre> public const string UrlTimeLine = https://weather.visualcrossing.com/VisualCrossingWebServices/rest/services/t imeline"; public static string WeatherApiKey = ""; private string GetWeather(string location, string dateTime) { if (string.IsNullOrEmpty(location) string.IsNullOrEmpty(dateTime)) { return "Sorry, I didn't get that. Please try asking in a different way"; } var restService = new RestService(); var url = \$"{UrlTimeLine}/{location}/{dateTime:yyyy-MM-ddTHH:mm:ss}? key={WeatherApiKey}"; var results = restService.GetWeatherData(url); if (results != null) { var description = results.Days[0].Description; var temperature = results.Days[0].Temp; return \$"{description}. The temperature of {temperature}°F"; } return "Sorry, I didn't get that. Please try asking in a different way"; } </pre> https://www.visualcrossing.com/resources/documentation/weather-data-tutorials/how-to-build-an-artificially-intelligent-chatbot-using-the-microsoft-bot-framework-and-the-visual-crossing-weather-api/ (emphasis added)
[1.7] invoking the selected domain agent to	Azure AI and Azure OpenAI meet this element.

Element	Analysis for Azure AI and Azure OpenAI
process the formulated request; and	On information and belief, Azure AI's orchestration relates to selecting the right skill or bot to handle user input. What is orchestration workflow? Article • 10/21/2024 • 5 contributors Feedback In this article - Example usage scenarios - Project development lifecycle - Reference documentation and code samples - Responsible AI - Next steps Orchestration workflow is one of the features offered by Azure AI Language. It is a cloud-based API service that applies machine-learning intelligence to enable you to build orchestration models to connect Conversational Language Understanding (CLU), Question Answering projects and LUIS applications. By creating an orchestration workflow, developers can iteratively tag utterances, train and evaluate model performance before making it available for consumption. To simplify building and customizing your model, the service offers a custom web portal that can be accessed through the Language studio. You can easily get started with the service by following the steps in this quickstart. Example usage scenarios Orchestration workflow can be used in multiple scenarios across a variety of industries. Some examples are: Enterprise chat bot In a large corporation, an enterprise chat bot may handle a variety of employee affairs. It may be able to handle frequently asked questions served by a custom question answering knowledge base, a calendar specific skill served by conversational language understanding, and an interview feedback skill served by LUIS. The bot needs to be able to appropriately route incoming requests to the correct service. Orchestration workflow allows you to connect those skills to one project that handles the routing of incoming requests appropriately to power the enterprise bot. https://learn.microsoft.com/en-us/azure/ai-services/language-service/orchestration-workflow/overview

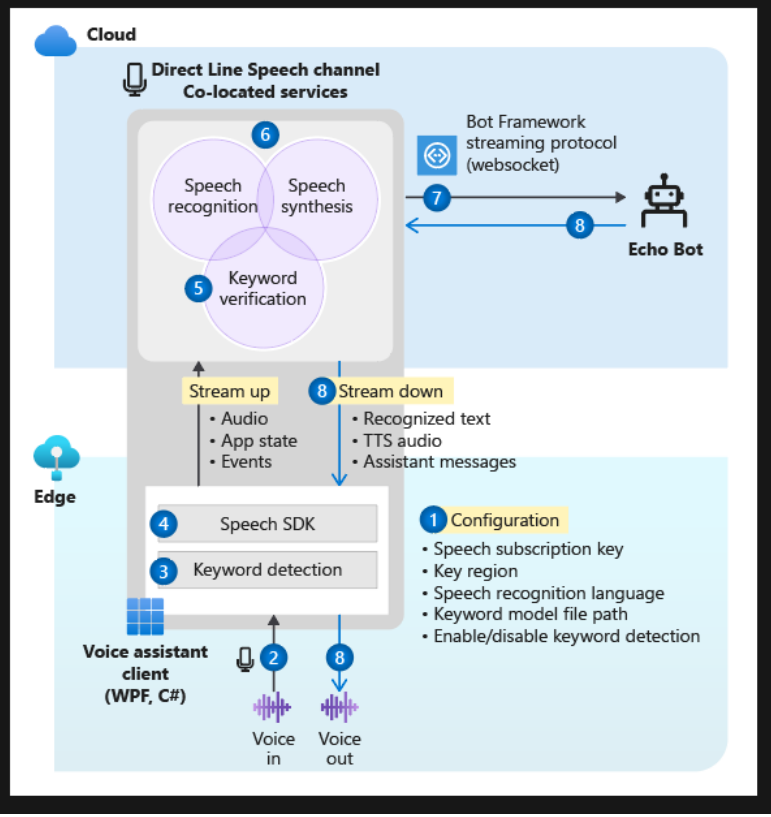
Element	Analysis for Azure AI and Azure OpenAI
	Use multiple LUIS and QnA models with Orchestrator If a bot uses multiple Language Understanding (LUIS) models and QnA Maker knowledge bases, you can use Bot Framework Orchestrator to determine which LUIS model or QnA Maker knowledge base best matches the user input. You can use the <code>bf orchestrator</code> CLI command to create an Orchestrator snapshot file, then use the snapshot file to route user input to the correct model at run time. This article describes how to use an <i>existing</i> QnA Maker knowledge base with Orchestrator. • For new bots, consider using the question answering and orchestration workflow features of Azure AI Language. • For more information about Orchestrator, see Intent recognition with Orchestrator in Composer. • For more information about the <code>bf orchestrator</code> command, see the Bot Framework CLI README. https://learn.microsoft.com/en-us/azure/bot-service/bot-builder-tutorial-orchestrator?view=azure-bot-service-4.0&tabs=cs

Element	Analysis for Azure AI and Azure OpenAI
	About this sample This sample is based on a predefined set of LUIS and QnA Maker projects. However, to use QnA Maker in your bot, you need an existing knowledge base in the QnA Maker portal. Your bot then can use the knowledge base to answer the user's questions. For new bot development, consider using Copilot Studio. If you need to create a new knowledge base for a Bot Framework SDK bot, see the following Azure AI services articles: • What is question answering? • Create an FAQ bot • Azure Cognitive Language Services Question Answering client library for .NET C# JavaScript  <pre> graph TD OnMessageActivityAsync[OnMessageActivityAsync] --> DispatchToTopIntentAsync[DispatchToTopIntentAsync] DispatchToTopIntentAsync --> ProcessSampleQnAAsync[ProcessSampleQnAAsync] DispatchToTopIntentAsync --> ProcessWeatherAsync[ProcessWeatherAsync] DispatchToTopIntentAsync --> ProcessHomeAutomationAsync[ProcessHomeAutomationAsync] </pre> <code>OnMessageActivityAsync</code> is called for each user input received. This module finds the top scoring user intent and passes that result on to <code>DispatchToTopIntentAsync</code>. <code>DispatchToTopIntentAsync</code>, in turn, calls the appropriate app handler. • <code>ProcessSampleQnAAsync</code> - for bot FAQ questions. • <code>ProcessWeatherAsync</code> - for weather queries. • <code>ProcessHomeAutomationAsync</code> - for home lighting commands. The handler calls the LUIS or QnA Maker service and returns the generated result back to the user.

Element	Analysis for Azure AI and Azure OpenAI
	Call the services from your bot For each input from your user, the bot logic passes in user input to Orchestrator Recognizer, finds the top returned intent, and uses that information to call the appropriate service for the input. C# JavaScript In the <code>DispatchBot.cs</code> file whenever the <code>OnMessageActivityAsync</code> method is called, we check the incoming user message and get the top intent from Orchestrator Recognizer. We then pass the <code>topIntent</code> and <code>recognizerResult</code> on to the correct method to call the service and return the result. <code>bots\DispatchBot.cs</code> C# (archived) https://learn.microsoft.com/en-us/azure/bot-service/bot-builder-tutorial-orchestrator?view=azure-bot-service-4.0&tabs=cs
[1.8] presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.	Azure AI and Azure OpenAI meet this element. For example, an Azure AI bot can present the result of user query in the text form.

Element	Analysis for Azure AI and Azure OpenAI
	5. At the chat prompt of Type your message, enter: How do I setup my surface book? The chat bot responds with an answer from your project.  https://learn.microsoft.com/en-us/azure/ai-services/language-service/question-answering/tutorials/bot-service For example, an Azure AI bot can present the result of a user query in the speech form.

Element	Analysis for Azure AI and Azure OpenAI
	The voice-enabled chat bot that you make in this tutorial follows these steps: 1. The sample client application is configured to connect to the Direct Line Speech channel and the echo bot. 2. When the user presses a button, voice audio streams from the microphone. Or audio is continuously recorded when a custom keyword is used. 3. If a custom keyword is used, keyword detection happens on the local device, gating audio streaming to the cloud. 4. The sample client application uses the Speech SDK to connect to the Direct Line Speech channel and stream audio. 5. Optionally, higher-accuracy keyword verification happens on the service. 6. The audio is passed to the speech recognition service and transcribed to text. 7. The recognized text is passed to the echo bot as a Bot Framework activity. 8. The response text is turned into audio by the text to speech service, and streamed back to the client application for playback.

Element	Analysis for Azure AI and Azure OpenAI
	 The diagram illustrates the architecture of the Azure AI Speech SDK, divided into Cloud and Edge components. Cloud Component: Direct Line Speech channel Co-located services: Contains three overlapping circles: Speech recognition (labeled 6), Speech synthesis (labeled 6), and Keyword verification (labeled 5). Bot Framework streaming protocol (websocket): Connects the Cloud services to the Echo Bot. Stream up: Carries Audio, App state, and Events from the Edge to the Cloud. Stream down: Carries Recognized text, TTS audio, and Assistant messages from the Cloud to the Edge. Edge Component: Voice assistant client (WPF, C#): The application layer on the Edge. Speech SDK: Labeled 4, it interfaces with the client and the cloud services. Keyword detection: Labeled 3, it handles initial voice input. Voice in/Voice out: Represented by microphone and speaker icons at the bottom. Configuration (labeled 1): - Speech subscription key - Key region - Speech recognition language - Keyword model file path - Enable/disable keyword detection Numbered steps in the diagram: 1 (Configuration), 2 (Voice in), 3 (Keyword detection), 4 (Speech SDK), 5 (Keyword verification), 6 (Speech recognition/synthesis), 7 (Bot Framework streaming protocol), 8 (Echo Bot). https://learn.microsoft.com/en-us/azure/ai-services/speech-service/tutorial-voice-enable-your-bot-speech-sdk