



VICA BEST PRACTICE

Chatbot Design and Training

By Virtual Assistant Team
[Email address]

Document Control

Version:	Update History	Release Date
1.0		

Table of Contents

1. Terminology	3
2. Training Best Practices	4
2.1 General	4
2.1.1 All intents should have distinct meanings	4
2.1.2 All intent variations should match the meaning/semantics of the parent intent	4
2.1.3 No duplicate variations should exist	4
2.1.4 Minimum number of variations	5
2.1.5 Number of variations should be balanced	5
2.1.6 Intent vocabulary should be balanced	5
2.1.7 Variations should be diverse	5
2.1.8 Variations should use correct spelling	6
2.1.9 Intents should have a single goal	6
2.1.10 Variations should use everyday language	6
2.1.11 Don't use intros and outros	6
2.1.12 Merge intents with a similar goal and meaning	6
2.2 Entity-specific Training	6
2.2.1 Use Entities instead of multiple intents	6
2.2.2 Entity annotations should be consistent	7
2.2.3 Annotated text in variations should have variety	7
2.2.4 Each entity should be used in many variations	7
3. Design Best Practices	8
3.1 Greetings and goodbyes	8
3.1.1 Welcome messages should let the user know about the bot's capabilities	8
3.1.2 Agents should have a suitable exit message when a session has ended	8
3.2 Helpful conversational features	8
3.2.1 Dialogs should support contextual requests	8
3.2.2 Dialogs should be flexible in collecting entity information	8
3.2.3 Confirmations with specified entities should be used for transactions	9
3.2.4 Ask clear and specific questions	9
3.2.4 Support small talk	9
3.2.5 Make use of rich media	10
3.2.6 Grice's Cooperative Principle	10
3.3 Conversation recovery	10
3.3.1 Recovery prompts after dialog interruptions	11
3.3.2 Agents should have customized fallback responses	11
3.3.3 Escalate the level of support if necessary	11
3.4 Persona	11
3.4.1 Responses should have a consistent style and tone that fits your brand	11
3.4.2 Avoid stereotyping	11
3.5 Testing	12
3.5.1 Test your app thoroughly with someone not involved in its development	12
3.5.2 Test your app on all platforms you plan on supporting	12

1. Terminology

Term	Definition
Annotations	The process by which content managers add and classify real-world user utterances into the chatbot knowledge base. In FlexAnswer, this is referred to as Teaching.
Channel	The medium by which the chatbot is communicating with the user. For example, desktop web, mobile web, Facebook Messenger, etc.
Entity	Term in an utterance that could be treated as a variable parameter. They could be nouns that are time, places, people, organizations, products, brands, etc. For example, in the utterance “What is the weather forecast for Singapore this Sunday?”, the term “Singapore” could be mapped to a Country entity, and the term “Sunday” could be mapped to a Date entity.
Fallback response	Chatbot response when it doesn’t understand the user utterance
Intent	The intention of the user’s utterance. In FlexAnswer, this is referred to as Master Question.
Named Entity Recognition (NER)	An NLP technique to identify named entities in the user utterance. Many NLP Engines (like Google Dialogflow) support NER.
NLP	Natural Language Processing NLP performance refers to how accurately the chatbot is able to predict user intents based on their utterances
NLP Engine	The NLP platform that the chatbot is developed on. For example, FlexAnswer and Dialogflow are NLP Engines.
Rich Media	Any form of chatbot content beyond simple text. For example, pictures, videos, buttons, cards, carousels, etc.
Session	The set of interactions and period of time from when a user starts engaging with a chatbot to when they disengage with it
Slot Filling	The chatbot may need to prompt the user to specify entity values required to complete that intent. For example, “Which country are you requesting the weather forecast for?” This is known as Slot Filling.
Transaction	A chatbot intent that involves a data change, and is not purely informational. Eg. signing up for a course, making a bank balance transfer.
Variation	Example utterance or phrase for a given intent. In FlexAnswer, this is referred to as Alternate Question.

2. Training Best Practices

The following are guidelines on how to create, manage and tune chatbot knowledge base content to optimize machine learning and NLP performance. Maintaining a high quality of content is essential to well-performing chatbots.

2.1 General

2.1.1 All intents should have distinct meanings

All intents (and associated variations) should have meanings that are distinct from other intents (and associated variations). In other words, different intents shouldn't contain similar variations, because this will prevent the NLP Engine from learning how to recognize those phrases.

If there are multiple intents with identical meanings, they should be merged.

DON'T	DO (Merged Intent)
Intent – How much is the LPA?	Intent – How much is the LPA?
Variation - Can you tell me how much the LPA costs?	Variation - Can you tell me how much the LPA costs?
Intent – What are the charges for an LPA?	Variation – What are the charges for an LPA?
Variation - What are the LPA fees?	Variation - What are the LPA fees?

2.1.2 All intent variations should match the meaning/semantics of the parent intent
 All intent variations should match the meaning of their parent intent. If a particular variation is too general, more details should be added to it as needed.

Sample Intent: “Golden Jubilee certificate registration”	
DON'T Variations	DO Variations
Golden Jubilee	How can couples register for the Golden Jubilee wedding certificate?
50 th Wedding Anniversary	I'd like to sign up for the 50 th wedding anniversary certificate

2.1.3 No duplicate variations should exist

There should never be duplicate variations, either between different intents, or within the same intent.

To determine how to fix duplicate variations between different intents, check on the following:

- Does the variation match the meaning of its parent intent?
 - o If not, you could consider adding more detail to the variation or deleting it, as appropriate.
- Do the parent intents have distinct meanings?

- If not, you could consider merging the intents.

2.1.4 Minimum number of variations

As a general guideline, to establish a minimum standard of NLP Performance, newly created intents should have a minimum of 10 variations for Google Dialogflow, and 20 for FlexAnswer and other NLP Engines.

In general, the more variations per intent, the better the NLP performance for that intent, although balance and diversity also have to be kept in mind. See the guidelines that follow for more details.

To aid in generating the minimum number of variations for each intent, crowdsourcing services like Amazon Mechanical Turk could be leveraged:

- <https://www.mturk.com/>

2.1.5 Number of variations should be balanced

The number of variations between different intents should be well-balanced, ie. intents should have a roughly similar number of variations. Otherwise, NLP classification will skew in favor of intents with a much larger number of variations. This means that care should be taken during the annotations process – real-world user utterances should only be added to an intent if they improve quality and diversity.

2.1.6 Intent vocabulary should be balanced

When adding new word combinations or synonyms to variations in an intent, also add similar vocabulary to all intents where those word combinations or synonyms could be used.

For example, for a banking chatbot that supports blocking a credit card, requesting the balance on the card, requesting a new card, etc., if the terms “visa card” and “gold card” are added to one of the intents, they should also be added to the others.

2.1.7 Variations should be diverse

While retaining the same meaning, include diverse variations of questions, commands, verbs, and synonyms for common nouns to ensure your phrases cover a broad spectrum of possible requests. Variations that are too similar to each other are not helpful to NLP performance.

Sample Intent: “Book a train ticket”	
DON'T Variations	DO Variations
Book a train ticket	Book train ticket
Book train	I would like to book a train ticket
Book ticket	I need to take the train
Book me a ticket	Can I order a train ticket
Train ticket	I need to be in Paris next Friday

	I want to travel to Brussels
	Please assist me in booking a boarding pass

2.1.8 Variations should use correct spelling

Care should be taken to ensure variations contain correctly spelled words. If incorrectly spelled words are encountered during the annotations process, they should be corrected when the utterance is added into the knowledge base. Use of local slang is acceptable if you expect it to be used in real-world utterances.

2.1.9 Intents should have a single goal

All parent intents and variations should express a single intention or goal. There should not be a case where multiple intentions are expressed – such cases should be handled by conversational dialog design.

In the same vein, all intent variations should not be overly long – if a variation takes up a paragraph or more, find a way to shorten it.

2.1.10 Variations should use everyday language

Variations should capture how users will ask questions or express an intent using normal, everyday language. Try to avoid expressions that are too formal or use industry jargon, if they are not likely to be used by normal users.

2.1.11 Don't use intros and outros

Don't use intros and outros as they are not relevant to the intent's meaning. For example, don't add the expression "Hello, I want to book a train ticket . Can I do that? Thanks". Instead, just add "I want to book a train ticket".

2.1.12 Merge intents with a similar goal and meaning

Don't create separate intents for user messages that are distinct in their meaning but have the same goal as response .

For example, if a user asks a chatbot 'I need more information about how to block a visa card', the user probably wants to block his visa card . So, don't create separate intents for 'blocking a credit card' and 'Info about how to block a credit card' as they will overlap .

2.2 Entity-specific Training

The following guidelines apply only to NLP engines that support Named Entity Recognition.

2.2.1 Use Entities instead of multiple intents

For example, if you want to have an entertainment chatbot that handles digital content such as watching TV series and movies, you can choose to have these

subject areas as separate intents, or the same intent with entities for specific data inside the sentences. In this case, the latter should be chosen for improved NLP performance.

The user utterances “Watch a tv series” and “Watch a movie” use the same vocabulary/verb of “Watch a “. The format is the same, so it should be created as a single intent with a “Content Type” entity distinguishing TV series vs Movie.

2.2.2 Entity annotations should be consistent

- Review your variations and ensure that entity annotations are pointing to the correct entities
- You should not have text in variations that is entity-annotated in some cases but not others.
- The span of text selected for an entity annotation should include all of, and no more than, the text that is necessary to match an entity.
- Be sure that the entity-annotated text in multiple variations contains similar portions of the variation. For example, consider that you have a variation "Set alarm at 6 a.m.", where "6 a.m." is annotated as [Date entity]. If you have another variation "wake me up at 7 a.m.", do annotate "7 a.m.", but do not annotate "up at 7 a.m."

2.2.3 Annotated text in variations should have variety

For example, if you are providing time values that should be parsed as [Time] entities in variations, do not provide the same time format in all variations. Your variations should have a variety of time examples like: "7 a.m.", "8 p.m.", "9 o'clock".

2.2.4 Each entity should be used in many variations

As a rule, each entity should be used in at least 5 variations

3. Design Best Practices

The following are guidelines on how to design chatbot knowledge base content to be more user-friendly and help the user to achieve their objectives in interacting with the chatbot.

3.1 Greetings and goodbyes

3.1.1 Welcome messages should let the user know about the bot's capabilities

Your agent's welcome intent should inform the user of top tasks the agent can help with, as well as brief descriptions (as needed) of how to use these features. This could be a list of the most common questions, for example.

3.1.2 Agents should have a suitable exit message when a session has ended

When a user completes a task in your agent, it should summarize the transaction/task and say something like "Until next time", etc.

3.2 Helpful conversational features

3.2.1 Dialogs should support contextual requests

This only applies to NLP engines that support contextual dialog. For example, if your chatbot handles requests for the weather and a user asks "What is the weather in Singapore today?", consider to support further requests like "How about tomorrow?".

3.2.2 Dialogs should be flexible in collecting entity information

This applies to NLP engines that support entities. For intents that require entity collection, your dialog design should support the user specifying some or all of the entities up front (Named Entity Recognition), and also support prompting the user for the remaining unspecified entities (Slot Filling).

	Example 1	Example 2	Example 3
User	"What is the weather forecast for Singapore this Sunday?"	"What is the weather forecast for Singapore?"	"What is the weather forecast?"
Chatbot	"Scattered showers in several areas"	"Which day are you requesting the forecast for?"	"Which day are you requesting the forecast for?"
User		"Sunday"	"Sunday"
Chatbot		"Scattered showers in several areas"	"Which country are you requesting the forecast for?"
User			"Singapore"
Chatbot			"Scattered showers in several areas"

3.2.3 Confirmations with specified entities should be used for transactions

When a user is conducting a transaction like placing an order or changing information, your agent should repeat what's happening for confirmation purposes. When creating these confirmation responses, make sure to include all user-specified entity values.

	Example
User	"I'd like to transfer \$100 from account 1234 to account 5678"
Chatbot	"Please confirm that you wish to transfer \$100 from account 1234 to account 5678"
User	"Actually, I want to transfer \$200"
Chatbot	"OK, please confirm that you wish to transfer \$200 from account 1234 to account 5678"
User	"Yes confirmed"
Chatbot	"Your transfer of \$200 from account 1234 to account 5678 has been completed. Transaction ID is 91011"

3.2.4 Ask clear and specific questions

Don't leave room for interpretation and make the success path more robust by using well-crafted prompts . Tell the user exactly what you want . A question might seem straightforward, but people could say anything . You should avoid leaving room for ambiguity.

	DON'T	DO
Chatbot	"Where do you live?"	"What is your home address?"
User	"At my parents place"	"10 Sunrise Lane"
Chatbot	"Sorry I didn't get that"	
User	"East Coast area"	
Chatbot	"Sorry I didn't get that"	

3.2.4 Support small talk

Support more personal interactions like "What is your name?", "What can you do?", "I need help", to establish a more human-like persona and establish trust with the user.

3.2.5 Make use of rich media

Support more personal interactions like “What is your name?”, “What can you do?”, “I need help”, to establish a more human-like persona and establish trust with the user.


3.2.6 Grice’s Cooperative Principle


Efficient communication relies on the assumption that there is an undercurrent of cooperation between conversational participants .

The flow of conversation needs to be natural and one that offers information that can help users get the answers to their questions and make their lives more simple .

Grice’s cooperative principle is a set of norms that are expected in conversations . It consists of four maxims, which we have to follow in order to be cooperative and for being understood:


Maxim of quality: As a speaker we have to tell the truth or something that is provable by adequate evidence. Do not advertise things you cannot live up to, such as saying, “How can I help you?” when really all the chatbot can do is take flight reservations.

How can I help you? 

Where would you like to fly to? 

When a chatbot is only designed for booking flights.


Maxim of quantity: We have to be as informative as required, we should not say more or less. Do not say “it’s 9:30 at night, Greenwich mean Time, 20 May 2019” if the user is asking “Do you have the time”. Just reply “Yes, it’s 9:30”.


It’s 9:30 at night, Greenwich mean Time, 20 May 2019 

Yes, it’s 9:30. 

When a customer asks for the time.


Maxim of relevance: Our response has to be relevant to the topic of discussion. For example, do not explain a return policy before someone has even placed an order.


Our return policy is 30 days. 

Sure, click here for a technical overview. 

When a customer asks for technical product specs.

Maxim of manner: We have to avoid ambiguity or obscurity; we should be direct and straightforward. Don’t use technical jargon that confuses the user

As stated in the civil code, article Art. 1641. 

Our warranty is 1 year in case of production errors. 

When a customer asks for the warranty policy.

3.3 Conversation recovery

3.3.1 Recovery prompts after dialog interruptions

For example, if the initial chatbot prompt is "What color do you want?" and the user replies with "jungle parrot", the chatbot should handle the interruption if possible, and then prompt the user to resume the conversation, eg. "Sorry, I'm not sure what "jungle parrot" is", "What color do you want?"

3.3.2 Agents should have customized fallback responses

When a user says something that isn't matched to an intent, the default fallback response should be customized to reflect your chatbot's branding, as well as provide information to guide the user to make a valid request.

	Example
User	"I'd like to transfer \$100 from account 1234 to account 5678"
Chatbot	"I'm sorry, but there is an issue with your account numbers. Your account numbers should be 9 digits long." "You can find your account numbers under the "My Accounts" section of your Internet Banking home page. Here's an example:" [Attached screenshot]

3.3.3 Escalate the level of support if necessary

If you detect that the user is having trouble fulfilling their intent (eg. through multiple consecutive fallback responses) or is expressing frustration (eg. through negative sentiment detection) you could escalate to a higher-level support channel (if available), such as live agent chat or a telephone support hotline.

3.4 Persona

3.4.1 Responses should have a consistent style and tone that fits your brand

As your users converse with your chatbot, it should feel like they're speaking to one persona. Make sure the qualities and personality you've chosen are represented in all of your responses.

3.4.2 Avoid stereotyping

Agents should be sensitive about cultures, genders, religious beliefs, abilities and ages. Stereotyping may offend users, even in jokes, and they might not return to your agent.

3.5 Testing

3.5.1 Test your app thoroughly with someone not involved in its development
Having someone unfamiliar with the agent use the app will give you insight to how naturally the conversation flows. Have them look out for accuracy, long pauses, missing conversational paths, pacing, awkward transitions, etc.

3.5.2 Test your app on all platforms you plan on supporting
If your agent will be available on one or more channels, make sure rich media and responses show up as expected across all platforms