

CHATBOT AND DIGITAL ASSISTANT TIPS, CONSIDERATIONS AND BEST PRACTICE

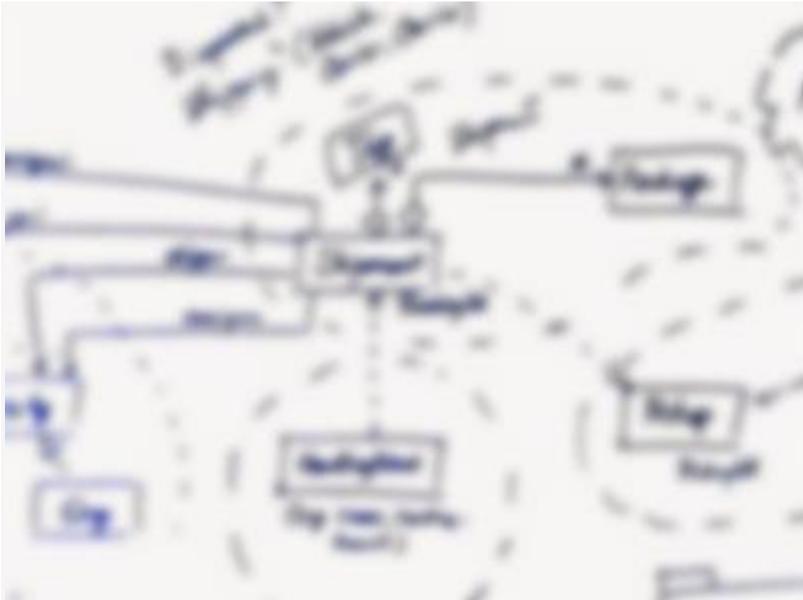


Stephen Garside

Starting a **chatbot or digital assistant project** can be a little daunting, especially if you have not done anything like it previously. Chatbots and digital assistants such as Google Assistant and Alexa can have lots of component parts, so I have compiled this list of **chatbot tips, considerations and best practices** to help get you started in your project.

Here is what we will cover in this article:

- Project Scope - What is the footprint of your chatbot.
- NLP Engine - Choosing a **Natural Language Processing** (NLP) engine.
- Knowledge Store - Where should you store the knowledge for your chatbot or digital assistant.
- Design Conversations - Identify and design your conversations.
- Persona - What is the personality of your digital assistant / chatbot.
- Memory - What should your chatbot remember from its conversations.
- Triggers and Redirects - Conversation flow and control.
- Context - What is it, why is it important, and how to control it.
- Slot Filling - Gathering information from within a conversation to provide the right answer.
- The Happy Path - Use 'discovery chips' to get users back on track.
- Feedback - Encourage simple and specific feedback on digital assistant / chatbot answers.
- Escalations - Options for keeping humans in the loop.
- Chatbot Analytics - Keeping track of usage.
- Conversational Interfaces - **Chatbot User Interface UI design considerations and best practice** tips.
- Call To Actions - Its all about the conversion funnel.
- Its Always Broken - Language models are never complete, why you should learn to accept it.
- Advances in Chatbot Tech - What does the future of Chatbots and Digital Assistants look like.



PROJECT SCOPE

The first thing to consider is the scope of your chatbot / digital assistant project. Do you just want a **basic website chatbot**, or a **digital assistant on Google Home, Alexa or Bixby**? Perhaps you are thinking enterprise level, so one chatbot that covers your website, smart speakers and social channels such as Twitter and Facebook.

The reason the scope of your chatbot project is important is because it will determine your technical approach. For a simple **question and answer chatbot** / digital assistant that has no dynamic content, you can probably get away with using software such as [Microsoft QNA](#), [Google DialogFlow](#) or [AWS Alexa](#) and just hard coding your answers into these systems. Most of these options allow you to upload a simple spreadsheet containing your Q and A's and they do the rest.

For an enterprise level solution then you are going to have to use the above mentioned NLP engines connected to custom web apis / services, perhaps something like a [Business Brain](#). This 'enterprise' approach to chatbot design will allow you to expose information about products and services that are dynamic in nature.

CHOOSE AN NLP ENGINE

NLP stands for Natural Language Processing, which basically means converting what humans say into something a computer can understand. This is often referred to as trying to **understand user intent**.

There are lots of NLP engines out there, but the key players are Google DialogFlow, Microsoft QNA or AWS Alexa. If you are building an enterprise level chatbot or digital assistant then you will have to learn how to use all three. For a simple website chatbot then QNA or Dialogflow are choices I would recommend.

The good news is that they all work in roughly the same way, so once you understand the key concepts such as **deriving intent**, **slot filling** etc then moving between them becomes easier.



WHERE TO STORE YOUR KNOWLEDGE

Where to store the answers to your questions is in itself a key question. If you are just building a simple Q and A chatbot, or perhaps a smart speaker app without any dynamic content, then chances are you can get away with storing your answers in your NLP engine of choice.

This approach quickly falls down if you want to scale your chatbot / digital assistant to multi channels such as websites, smart speakers and social media. You don't want to have to keep the same knowledge up to date in multiple locations - it simply does not scale. Also, what happens when there is a new NLP kid on the block such as Bixby - do you load your knowledge into Samsung's NLP engine too?

If you are thinking enterprise, then you need to build a custom knowledge store, which is essentially a database of your answers to the questions posed via the NLP engines. At this point you should consider building a scalable business brain or knowledge framework.

The image shows a screenshot of the Dialogflow console interface. On the left is a sidebar with the Dialogflow logo and a list of navigation items: Intents, Entities, Knowledge [beta], Fulfillment, Integrations, Training, Validation [beta], History, and Analytics. The main area displays the configuration for a service named 'GateFittingService'. It includes sections for 'Contexts', 'Events', and 'Training phrases'. The 'Training phrases' section contains a list of user expressions, each preceded by a quote icon: 'Add user expression', 'Do you install gates', 'Can you fit the gate for me', 'Do you offer a fitting service', 'Who can fit my gate', and 'Do you offer a driveway gate fitting service'.

DESIGN YOUR CONVERSATIONS

Conversation design is a huge topic and beyond the scope of this article. I recommend reading articles such as the [google guidelines for conversation design](#), which explain best practice far better than I ever could.

Once you understand **conversation design best practice**, then the following tips will help you identify and create the conversations you want to have with your customers:

- Identify all the top level conversations you want to have with your customers - write them down in a list.
- Group your conversations into sub-groups i.e. Product Info, Aftercare, Smalltalk etc.
- Give each conversation a unique name, prefixed with the subgroup e.g. ProductInfo_Prices, Aftercare_Maintenance, SmallTalk_Introduction etc

Next, work through each conversation and:

- For each conversation, write down a brief summary of what you want the outcome to be.
- Write down as many examples as you can of what people will say to trigger the conversation - these are known as **User Utterances**.
- Identify the key words within the utterances that are required to answer the question i.e. Zipcode, Product Name, Email etc - these are known as **Slots**.
- If the user does not provide all the information in their utterance, write down examples of what your chatbot will ask to determine these slots e.g. What is your Zipcode? - this is known as **Slot Filling**.
- Most conversations branch or pivot. Identify these branches and work through each branch to identify extra slots.
- At the end of each branch in the conversation, either write down your answer, or where you will get the answer from i.e. the product database, or Q and A database. These become your **Answer Finders**.
- For each conversation branch, identify any opportunities or call to actions for the end of the conversation. Can you offer an extra service, or perhaps trigger another conversation.
- For each question, consider how your answer might differ on a smart speaker vs a chatbot window or smart display e.g. A customer asks for directions on a phone - display a map for a visual response, vs read out directions on a smart speaker.

CHATBOT / DIGITAL ASSISTANT PERSONA

Defining the personality (persona) of your digital assistant / chatbot is one of the first steps you should take. Does it have a name, is it going to be female, male or gender neutral. What age is your chatbot, what tone of voice does it use and does it use humor?

Answers to these questions will affect the language you use in your responses. Design your chatbot persona before you write the answers to your questions.



Chatbot Steve

“witty, funny, a charming bot!”

DEMOGRAPHICS

Age: 36
Location: Halifax, UK
Job: Digital Assistant
Interests: Helping Customers, Love Island

PERSONALITY TRAITS

Chatbot Steve is an extremely well educated and articulate kind of bot. A key skill is consistency, you can depend on this dude 24 / 7!

Steve is a really funny bot, but can be serious when needed. If you have an issue, then Steve has all the empathy you could ever ask of a bot.

Steve never swears, and is often seen sporting a halo.

MEMORY

There is nothing more annoying than being asked the same question over and over again. As an example, the first question a customer asks might require their postcode to provide an answer. Their second, unrelated question might also require a postcode, so what do you do? Most 'out of the box' chatbot frameworks will ask the same question twice - really annoying!

If you are creating an enterprise level chatbot then you need to consider coding in a 'memory' feature. It might be that you just remember certain slot values (parameters) such as postcode, address, customer name etc. This will allow you to skip certain steps within the conversation if you already have the slot values in memory.



CONVERSATION TRIGGERS AND REDIRECTS

Following on from the [Memory](#) issue outlined previously, you may find you need to progress users to other parts of the same conversation (for example if you already their postcode in memory). You may also want to trigger an entirely different conversation based on a slot value they have provided.

Coding in the ability to redirect to a different conversation entirely, or jump to a different step within the existing conversation is very useful. Luckily, NLP engines like Dialogflow provide the ability to [trigger custom events](#). It is possible to trigger a step within the current conversation, or a step in a completely different conversation.

KEEPING CONTEXT

Context in the world of a chatbot or digital assistant basically means the ability to associate user input to the current conversation. As an example:

Customer - "Can you send me a brochure"

Chatbot - "Sure, would you like a copy sending in the post?"

Customer - "Yes please"

Without Context, "Yes please" means nothing to the chatbot - what is the customer saying yes to? In the example above it has meaning within the context of the conversation. The trick is to end the current conversation context at the correct time otherwise the user will start to experience weirdness! DialogFlow allows full control over context, however not so with AWS Alexa - you will have to write your own context engine.



SLOT FILLING

How do I clean **XYZ** product

What time is your **Elland** store open until on **Tuesday**?

Can you send me your latest **desktop pc** brochure

What do you recommend I use to get **chewing gum** off my **carpet**?

My postcode is **HX7 1TZ**

Slot filling is the process of gathering the key pieces of information required to answer a question. e.g.

Customer - "Can you send me your brochure please?"

Digital Assistant - "Sure, what is your email address?"

In this example the 'slot' to be filled is the customer's email address.

NLP's such as DialogFlow and AWS Alexa both have the concept of slot filling, but handle the process differently. DialogFlow can handle it for you by asking a question on your behalf e.g. "Sure, what is your email address". AWS delegates slot filling entirely to your custom answer service, so be prepared to right your own logic!

I recommend you read more about the [key differences between DialogFlow and AWS Alexa NLPs](#).

KEEP USER ON THE HAPPY PATH

Without doubt, one of the most frustrating things about using a chatbot or digital assistant is **conversation discovery**, by that I mean how does the customer know what your chatbot can and can't do? Luckily there are several ways you can funnel users to your conversations if they are having trouble.

The first is to pre-condition them on how to get the best out of their interaction with your chatbot. Tell them up front to **keep sentences short** and **ask one question at a time**.

Next, you can tell them to ask 'Help' at any time to get a list of things your chatbot can help with.

One technique is to use **discovery chips** to guide users towards your key conversations. Discovery chips are simply buttons that act as conversation starters e.g.



Discovery chips can start other **discovery conversations** so you can sub-categorise conversations, in essence allowing users to drill down to the conversation they want to have. Don't fall into the temptation of starting out with discovery chips, only offer them when your customer has asked for help or your chatbot is unable to answer a question, otherwise your natural language experience becomes nothing more than a conversation tree - that's not NLP!

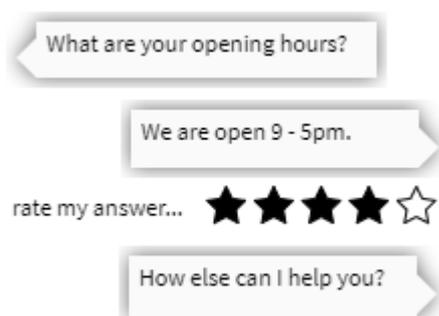
GATHERING USER FEEDBACK



One of the dangers of customer facing chatbots and digital assistants is gauging how satisfied a customer is with the answers a chatbot is providing on behalf of your business. There are several ways you can mitigate this risk:

Implement **sentiment analysis**, which is basically giving each customer / chatbot interaction a positive, neutral or negative score based on the language used by the customer. There are lots of sentiment analysis services out there, DialogFlow offer it as part of their NLP (at a cost), others include [Microsoft Text Analytics](#), [Google Sentiment Analysis](#) and a nice freebie called [Vader Sentiment](#).

Another way to gather user feedback is to create a feature that allows users to quickly give feedback - this should relate to a specific answer within a conversation so you understand exactly what answer is causing friction.



ESCALATION AND HUMAN INVOLVEMENT

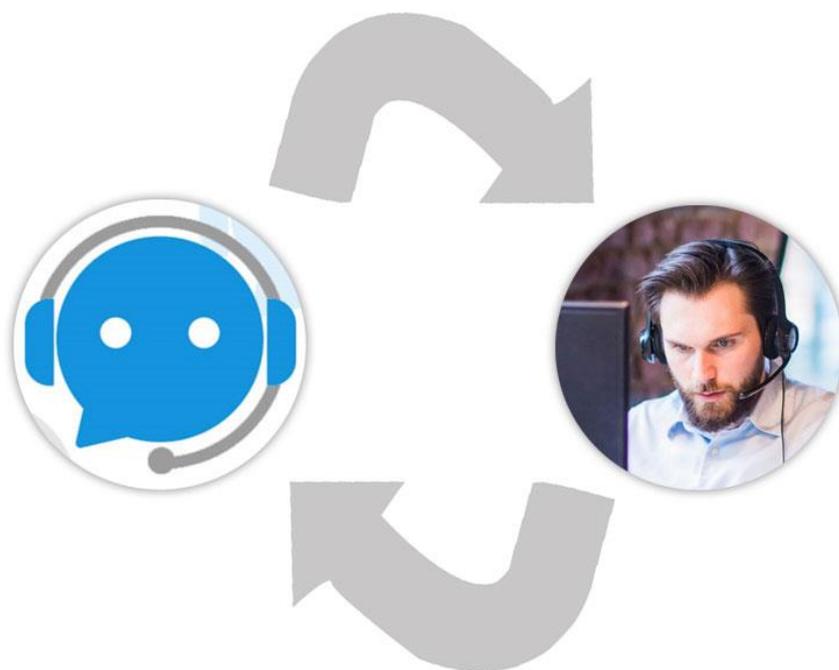
A key benefit of chatbots and digital assistants is the lack of human involvement in the conversation and the cost benefits this brings. That being said, current NLP technology has its limitations, and the last thing you want to do is destroy your customer service experience.

For chatbots, you have 3 options:

- Leave it all to the chatbot and carefully monitor interactions.
- Offer an escalation path by handing the user over to a Live Chat facility to talk to a human if sentiment is tanking.
- Implement a **blended chatbot interaction** where the conversation is automatically picked up by a human if the chatbot cannot answer a question.

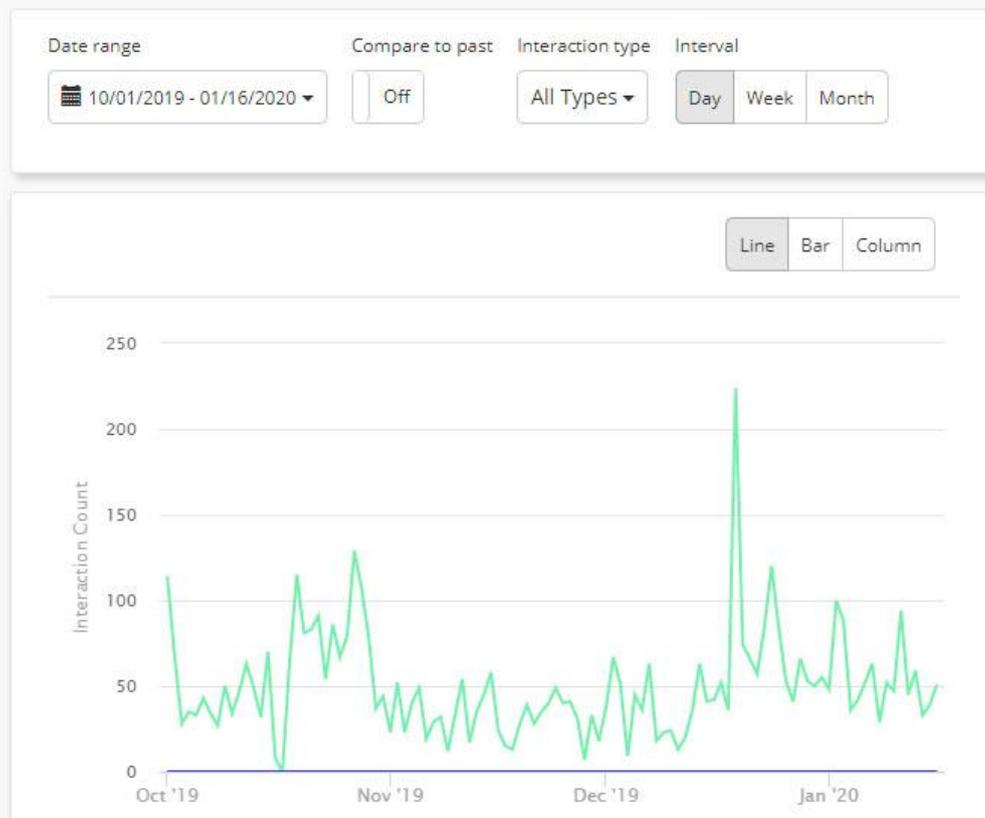
For digital assistants on smart speakers / displays and mobiles phones, a blended approach will not work, so your assistant may have to offer to call or email the customer back should sentiment analysis indicate a problem.

As NLP engines improve this should become less of an issue, but considering your approach to human involvement up front makes good sense.



CHATBOT ANALYTICS

Interaction Count ¹



So you launch your chatbot or digital assistant voice app on a smart speaker - then what? How do you know if it is being used, what questions are being asked and what is the demographic of your users? This is where **chatbot analytics** come in.

Most NLP engines provide their own analytics, which is fine if you are just building an Alexa or Google Home app. If you are creating an enterprise level solution, you should put all your analytics in one place, so you basically have 2 options:

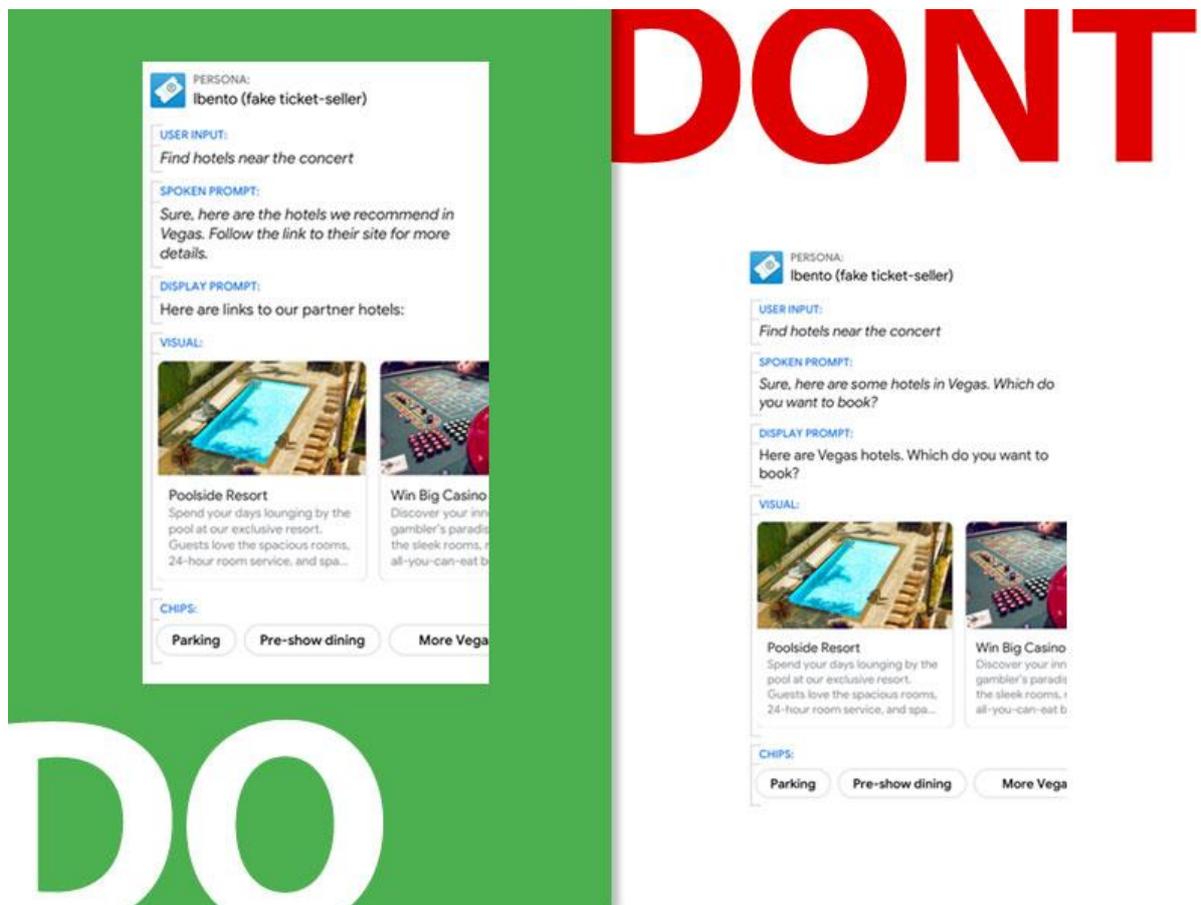
- Write your own.
- Use a product such as [Dashbot](#) that offers both a free or paid subscription for all the charts and analytics you could wish for. There is an element of integration, but its pretty straight forward to code.

CONVERSATIONAL INTERFACES

Designing conversational interfaces for chatbots and digital assistants is a huge topic beyond the scope of this article, however there are a few key principles to consider.

If you are building an enterprise level natural language experience across multiple channels such as web, phone and smart displays, then you want a consistent experience for your user. A great place to start for the types of controls you might want to incorporate into your chat experience are the [google guidelines on conversational interface visual components](#). Sticking to these basic controls in a website chatbot window will ensure that when your customer switches their conversation to a digital assistant on a phone or smart display they get a similar experience.

Whilst these basic controls may feel a little restrictive, components such as the [google interactive canvas](#) promise rich HTML interactions within a conversational environment. These same interactions can be served via an iframe in your website chatbot window ensuring consistency is maintained.



CALL TO ACTIONS

A key benefit of chatbots and digital assistants is conversation consistency, in particularly the application of **chatbot call to actions**.

One 'rule of thumb' worth following is to identify what you want the customer / user to do next when they reach the end of every conversation branch, and provide a means to do it. This can be as simple as asking a follow up question e.g ."Here is the product information you requested. Would you like me to send you a colour brochure?".

An alternative approach is to offer a set of choices using chips (buttons with actions). The key thing is - at the end of this conversation what is the next step and how do we encourage the user to take it?

A LANGUAGE MODEL IS ALWAYS BROKEN

The language model behind a chatbot / digital assistant application is forever broken. You will go live with a broken language model, and that will never change. It will improve over time as customers interact with it and you make tweaks based on what customers ask, but it will always be broken. Language models are never complete, they are always in a state of being fixed.

To prove the point, everyday between 16 and 19 percent of queries in google have never been asked before , so the next time someone criticises your chatbot for not being able to answer a question - quote google!

The key is to gracefully handle the fact that your chatbot does not have an answer. Offer discovery chips, or suggestions about what it can deal with. If appropriate, perhaps inject a bit of humor into the response, and re-assure the user that their unanswerable question has been logged and to try again in a couple of days once the Q/As have been updated. Always make sure you have an unanswered question review process in place.

CHATBOTS IN THE FUTURE

Current chatbot technology is smart, but not that smart. Most NLP frameworks rely on structured, pre-built conversations and language models hooked into some form of Q and A database or web service / api.

Next generation technology will incorporate artificial general intelligence techniques that will eradicate the need for pre-built conversations. The need to map example user utterances to intents will no longer be required, instead, intelligent software will handle requests real-time and mine information from a variety of sources, similar to how systems such as IBM Watson currently operate, but much smarter.

Smart displays and TVs will play more of a role in connecting customers to businesses. E-commerce and customer service interactions will become unique, voice driven experiences rather than the current one size fits all website experience.

Google's interactive canvas control demoed in this video example of a voice first eCommerce experience on a smart display shows how this type of technology will enable voice driven, visual apps to proliferate smart phones, smart displays and tvs.

For now, businesses investing in NLP, chatbot and digital assistant technologies will be able to quickly take advantage of these new mediums as they become mainstream. Getting your **business NLP ready** now will ultimately reap rewards in years to come!

I hope these **tips, considerations and best practices** will help you prepare for your organisation for your **chatbot or digital assistant project**.