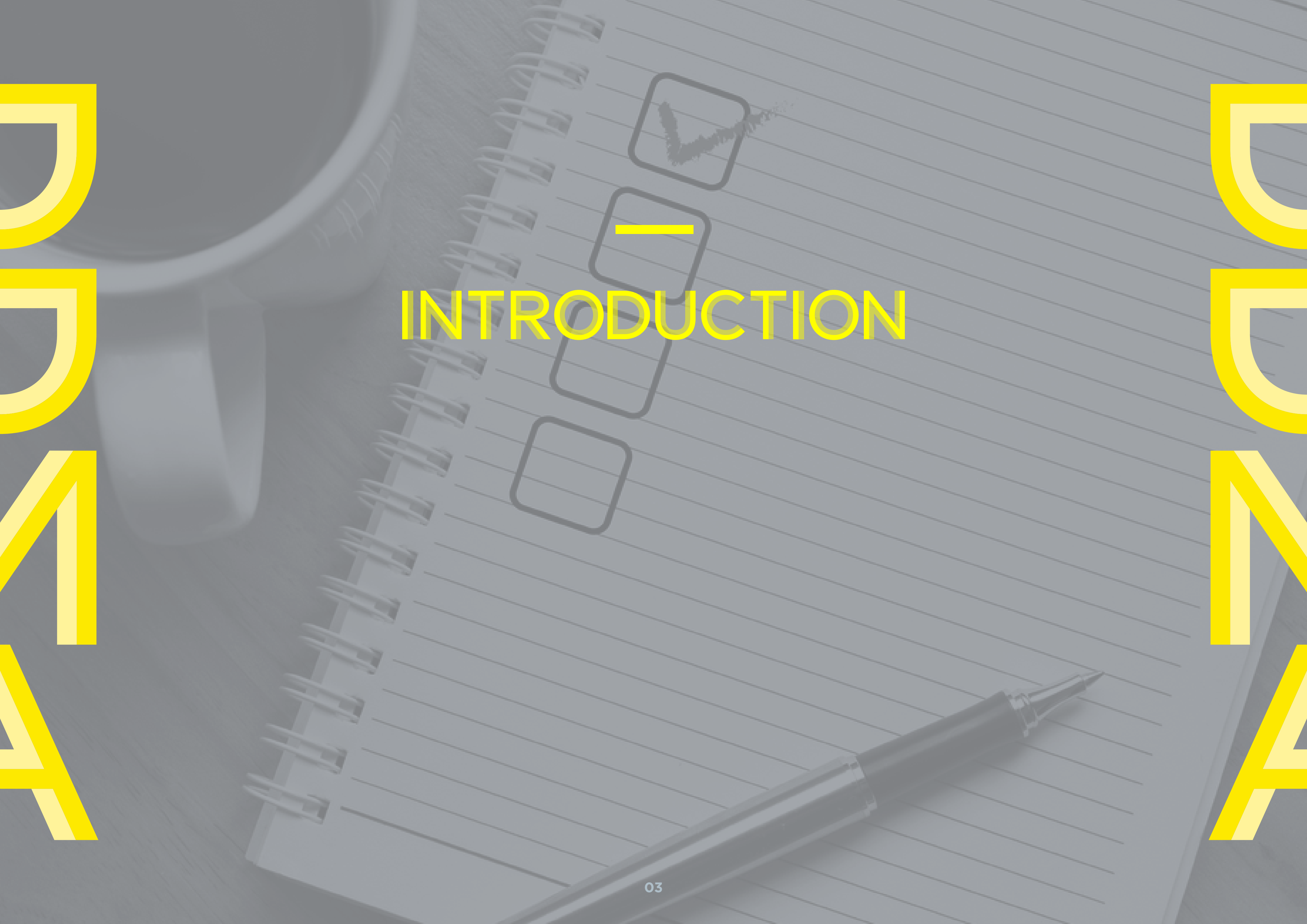
THE ULTIMATE GUIDE TO CONVERSATIONAL ANALYTICS

Choosing the right metrics for improving
the customer experience of your voice and chatbots

TABLE OF CONTENTS



INTRODUCTION

INTRODUCTION

It appears that most organisations employing chatbots and voicebots rarely use analytics. This is a missed opportunity since a well-designed and smart use of analytics can be crucial to improving the customer experience. How then, to set up your conversational analytics? That is exactly what we hope to explain to you in this whitepaper.

Working with voice or chatbots, how do you make sense of all the data you can collect? Either working with web-integrated chat applications, or applications built in well-known virtual assistants, the question on which data to collect and how and where to store it remains. And more importantly: how do you translate all your collected data into actual actions to improve your conversational implementations? In this whitepaper, we will guide you step-by-step through all the different kinds of data you can consider monitoring in your conversational analytics.

We focus on data that can be gathered under three different categories of metrics: *conversation-related metrics, chat session & funnel metrics and bot model health metrics*. After a brief description of all categories in chapter one, we will take a deep-dive into each category in the following chapters. The fifth and final chapter discusses the conversational analytics of Rabobank, a Dutch bank that uses a chat and voicebot to communicate with customers.

The DDMA Committee Voice
This whitepaper is written by Lee Boonstra, Applied AI Engineer and Developer Advocate at Google and Justin Meijer, Product Manager - Guide & Engage at Rabobank Nederland, both members of the Committee Voice of the Dutch Data-Driven Marketing Association (DDMA). As committee committed to voice, we believe voice is of great, if not essential value to the customer journey. We aim to increase belief in the "voice channel" by showing brands and publishers its worth for marketing, sales and service purposes. We strive to bring voice from the early adopter stage to the early majority stage and give it a permanent place in digital transformation processes.

As a committee, we have a voice-first vision, but we also look at multimodal developments and related channels, such as chat. We play a leading role in the development of voice by sharing knowledge and inspiring cases from home and abroad and laying out the latest features in voice technology.

We hope this whitepaper will help you properly organize your conversational analytics in order to make the most out of your chat or voice implementations. Good luck!

Maarten Lens-Fitzgerald
Voice Evangelist, Chairman Committee Voice

Daan Gönning
Business Director at Doop

Anja de Castro
Conversational UX Designer

Lee Boonstra
Applied AI Engineer and Developer Advocate at Google

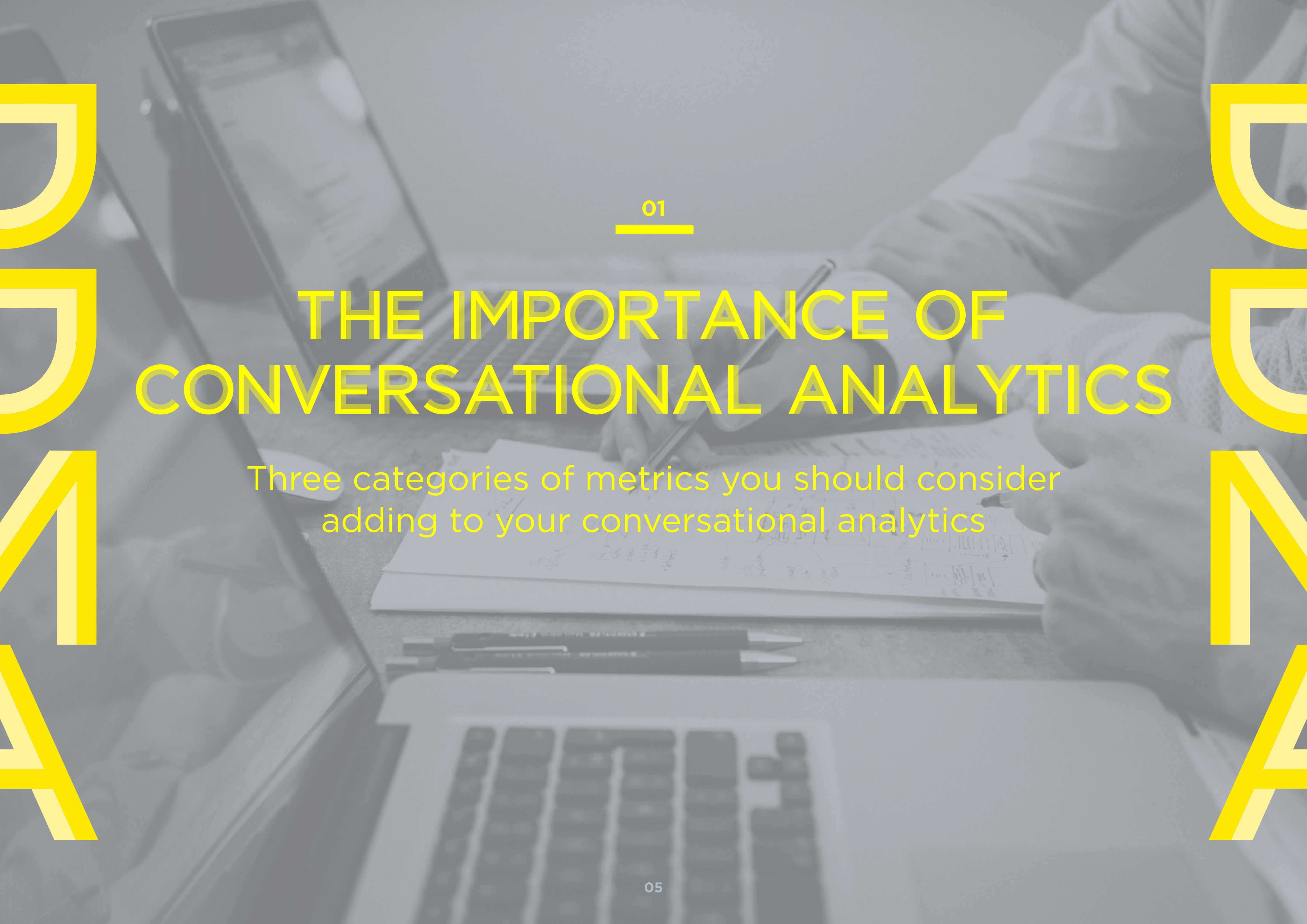
Lieneke Grollé
Innovation Lead at KRO-NCRV

Justin Meijer
Product Manager - Guide & Engage at Rabobank Nederland

Carla Verwijmeren
Founder at Smartvoices, Teacher at SRM

Richard de Vries
Conversational AI Lead at Philips

Marike van de Klomp
Specialist Channel Transformation & Strategy at Vattenfall Sales



01

THE IMPORTANCE OF CONVERSATIONAL ANALYTICS

Three categories of metrics you should consider adding to your conversational analytics

THE IMPORTANCE OF CONVERSATIONAL ANALYTICS

Three categories of metrics you should consider adding to your conversational analytics

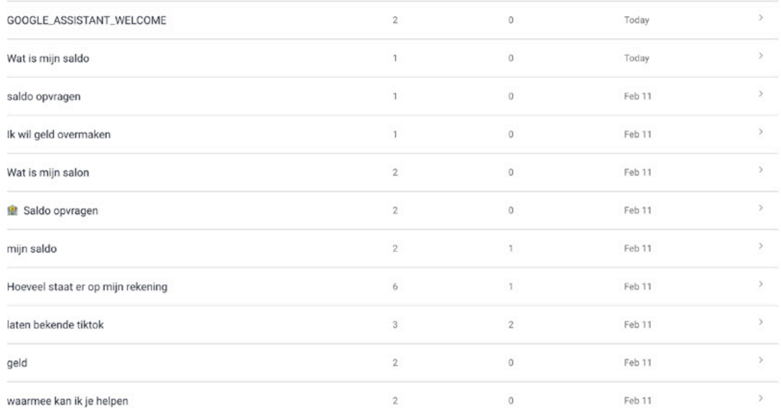
How then to make sense of all the data you can collect? What kinds of data typically lends itself best to conversational analytics? And why should we use conversational analytics in the first place? In this chapter we stress the importance of conversational analytics, as well as an introduction to three different kinds of metrics that you should consider to monitor with it.

We are all familiar with the problem; you are trying to have a conversation with a voicebot , but the virtual agent doesn't understand what you're saying. A default fallback message follows or worse, a wrong answer! On most occasions, the chatbot or voicebot is working fine, but it is trained to answer different types of questions, caused by the fact that organisations rarely integrate conversational analytics into their conversational design. This way, organisations don't understand how your users are using their virtual agent.

1.1 The importance of conversational analytics

An excellent conversational design would of course prevent the problems mentioned above from arising. According to Google's guidelines for conversational design, the solution lies in informing users what your assistant can and can't do. For example, an assistant has to welcome users during start-up and explain how it can help them. This way, you guide users to asking the right questions. To accomplish this, you have to have an idea of the problems the assistant is expected to solve for users. That's why conversational design should always be based on the data that tells you something about these problems. As an organisation working with chat or voice, you probably already have this data at your disposal. Is this your first attempt building a virtual agent? Then consider data deriving from other channels like contact centres, social media and e-mail, to retrieve all questions, topics, complaints that are relevant to customers that want to communicate with your organisation.

When building such a virtual assistant, it is essential to set up analytics directly after going live. Conversational analytics is not just a nice-to-have addition to your current data collection methods, it is crucial to improving the customer experience of your bots. Conversational analytics is what gives you direct feedback about the way customers interact with your bot. However, experience has taught us that organisations often overlook this valuable data. Our advice is: don't spend a year of endlessly improving conversations when you can learn so much from live traffic data. Even if you only collect data on a small selection of variables, you can use it to grow your bot and quickly make it smart. After that, by gradually adding more topics to your design based on this conversational data, you will be able to build a bot that has answers to any questions.



GOOGLE_ASSISTANT_WELCOME	2	0	Today	›
Wat is mijn saldo	1	0	Today	›
saldo opvragen	1	0	Feb 11	›
Ik wil geld overmaken	1	0	Feb 11	›
Wat is mijn salon	2	0	Feb 11	›
🏦 Saldo opvragen	2	0	Feb 11	›
mijn saldo	2	1	Feb 11	›
Hoeveel staat er op mijn rekening	6	1	Feb 11	›
laten bekende tiktok	3	2	Feb 11	›
geld	2	0	Feb 11	›
waarmee kan ik je helpen	2	0	Feb 11	›

Query's to Rabobank's voicebot

1.2 Three different categories of metrics

While setting up conversational analytics, there are three specific categories of metrics relevant to designing a voicebot : *conversation-related metrics, chat session & funnel metrics,* and *bot health metrics*.

1.2.1 Conversation-related metrics

Conversation-related metrics can help understanding conversations and shining a light on questions like: what's been said, by who, when and where? To effectively monitor *conversation-related metrics*, data could be stored in a *data warehouse*: an enormous database to which several data sources can be connected. Here, you can store as much structured data as you want, whether it's website data, website logs, login data, advertising data or Dialogflow chatbot conversations. The more data you gather, the better you can understand and help your customers.

The six most important conversation-related metrics needed to improve chatbot or voicebot conversations are as follows (we will delve deeper into conversation-related metrics in chapter 2):

1. A **session ID** to find and read transcripts of specific sessions and identify the number of individuals using your bot.
2. A **date and time stamp** to find and read transcripts within a specific timeframe and determine session length.
3. A **sentiment score** to find transcripts based on particular sentiments.
4. A **dialogue language** and a **keyword** to find all transcripts in a specific language or with certain words.
5. A **platform setting** to find all transcripts from a specific platform. This metric is particularly important with multichannel chat/voice applications.
6. **Intent Detection information**, like *detected intent name, fallback* or *end-of-interaction messages* to determine whether the bots answered correctly.

1.2.2 Chat session & funnel metrics

To obtain an idea of the chat funnel and therefore identify the structural course of conversations users have with your bot, it is important to incorporate chat session and funnel metrics into your conversational analytics. These metrics allow you to visualise the conversational route users take when they engage with your bot. Your conversational analytics could monitor the following eight chat session and funnel metrics (we shall explore this type of metrics in greater detail in chapter 3):

1. Total usage of people using your virtual agent
2. The percentage of users whose queries matched the correct intent, and the number of queries the intent was matched to.
3. Completion rate
4. Drop-off rate
5. Drop-off place
6. User retention
7. Endpoint Health, which indicates the extent to which different systems are correctly linked
8. Google Assistant: discovery information on how users came upon your Action

1.2.3 Bot Model Health metrics and tools

Working with well-known bot building tools, it's likely that your bot uses Natural Language Understanding – a form of machine learning – to understand (text) queries and match them to a specific *intent*, a process referred to as *intent classification*. If this process goes smoothly, you can be sure users will get the right answers to their questions. However, to ensure *intent classification* runs smoothly, there are ten Bot Model Health metrics and tools you could monitor (we shall explore these kinds of metrics and tools in greater detail in chapter 4):

1. A **true positive** indicates that the bot matches a query to the correct *intent*.
2. A **true negative** means that the bot matches a query to the correct *fallback message*.
3. A **false positive** indicates that the bot matches a query to the wrong *intent* and the wrong *fallback message*.
4. A **false negative** means that the bot wrongly matches a query to a fallback message, while the correct intent exists.
5. **Accuracy**: the ratio of correctly predicted observations to the total number of observations.
6. **Precision**: the ratio of positive prediction values
7. **Recall** and **fallout**: the *sensitivity ratio* and *false alarm ratio* respectfully
8. **F1 score**: the weighted average score of precision and recall

- **Confusion Matrix**: a table used to describe the performance of a classification model on a set of test data.
- **ROC curve**: A graphic representation indicating how well a model can distinguish intents.

02

UNDERSTANDING CONVERSATIONS

Six conversation-related metrics to track and store in a data warehouse

02

UNDERSTANDING CONVERSATIONS

Six conversation-related metrics to track and store in a data warehouse

Bot-building tools like Dialogflow have their own built-in analytics. Although they are convenient, to gain deeper insights into your chatbot or voicebot , it is advisable to gather and link your data together in a data warehouse. This chapter will explain the benefits of a *data warehouse* by using it as the storage location for your conversation-related data.

2.1 Why a data warehouse?

Using a *fully-managed, serverless data warehouse*, you can analyse enormous amounts of data in a scalable way. An example of this is Google's BigQuery data warehouse (available as Software as a Service), in which you can store a lot of data supported by ANSI SQL language.

Compared to built-in analytics in bot-building tools like Dialogflow, BigQuery analytics has a lot of legal benefits. With BigQuery, you can easily meet relevant legislation like the GDPR, since it allows you to:

1. Choose where data is stored.
2. Choose how long data is stored.
3. Create back-ups of your conversational data.
4. Remove sensitive *personally identifiable information (PII)* from your data before storing it. According to the GDPR, the storage of PII is not allowed.
5. Create custom-made data sources, such as chat translations or user sentiment.
6. Combine chatbot conversations with other valuable data to gain useful insights for creating and improving omnichannel customer experiences.

2.2 Which conversation-related metrics are important?

The more data you store in a data warehouse, the better you can understand and serve customers. If you store data correctly, you can gain insights categorised under several metrics. You should monitor six conversation-related metrics as a chatbot or voicebot builder: *a session ID, sentiment score, language & keyword, platform* and *intent detection information*.

1. Session ID
A session ID is needed to find transcripts of specific conversations users had with your bot and determine the total number of your voice application's unique users. Bot-building tools such as Dialogflow automatically allocate each chat session a *session path*, each one connected to your agent account. Each *session path* is unique because of its *universally unique identifier* (UUID):

```
const sessionId = uuid.v4();
const sessionClient = new df.SessionsClient();
const sessionPath = sessionClient.sessionPath(projectId, sessionId);
```

Storing your data in a data warehouse even allows you to categorise each individual utterance under a *session ID*. With a session ID, you can retrieve full chat transcripts, organised under the data fields under which you stored your data. For example, if you want to retrieve a session ranked by date and time, the SQL-query with the session ID would look like this:

```
SELECT * FROM `chat_msg_table` WHERE SESSION_ID = ‘projects/myagent/agent/
sessions/db33b345-663c-4867-8021-fecd50c5e8b1’ ORDER BY DATETIME
```

2. Date and time stamp

A date and time stamp is necessary to retrieve transcripts from sessions between a specific period and calculate session lengths. If your bot-building tool doesn't allow you to store data under a date and time stamp, you can set your own time set in a data warehouse. If you choose this, you have to make sure the data warehouse can read the date/timestamp information. In the case of BigQuery, a date and time stamp will look like this:

```
const timestamp = new Date().getTime()/1000;
```

Also, date and time stamps have to be stored under a *session ID*, together with the other metrics belonging to the same session ID. With the right SQL query, you can retrieve chat messages within a certain period, in chronological order. For example, if you want to retrieve messages sent between August 1 and 20, 2020, the SQL query will look like this:

```
SELECT * FROM `chat_msg_table` WHERE DATETIME > ‘2020-08-01 10:00:00’ AND
POSTED < ‘2020-08-10 00:00:00’ ORDER BY DATETIME DESC
```

3. Sentiment Score

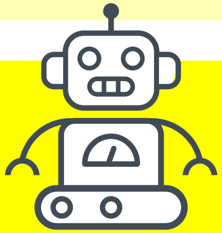
A *sentiment score* is necessary to retrieve transcripts based on specific sentiments. For instance, if you want to look up transcripts from sessions in which the customer made negative remarks about one of your products or services. Built-in bot-building tools such as Dialogflow already monitor *sentiment score* by default, only not for Dutch. Therefore, to do a sentiment analysis based on Dutch chat/voice conversations, these transcripts need to be translated first. Translating tools such as Cloud Translate en Cloud Natural Language could come in handy. This how you make a sentiment score readable for BigQuery:

```
const sentiment_score: queryTextSentiment.score;
const sentiment_magnitude: queryTextSentiment.magnitude;
```

The *score* indicates the dominant emotion within a document. The *magnitude* of a sentiment suggests how much of that emotion is present. In most cases, this value is proportional to the length of the document.

Using a data warehouse, you should store the sentiment score under a *session ID*, which will allow you to retrieve messages based on sentiment. For example, if you want to retrieve messages ordered by sentiment, the SQL query will look like this:

```
SELECT * FROM `chat_msg_table` WHERE SENTIMENT_SCORE < 0 ORDER BY SENTIMENT_
SCORE ASC
```



Tip: How to deal with sarcasm

A chatbot or voicebot takes everything users to say very literally. How do you deal with sarcasm then? Especially since sarcasm is hard to understand for even humans. Take, for instance, a remark as:

"Yay! Due to the snowstorm, my flight has been cancelled. How Great!"

The sentiment models of Google Cloud will probably perceive this remark as very positive because of the words: "Yay!" and "How great!" and rate it with maybe even a 90% (0.9) sentiment score. Unfortunately, we want to classify this as very negative, because we are talking about cancelled flights which, obviously, are not fun. That's why we expect a negative sentiment score.

It's nice to know that you can train your sentiment models to recognise sarcasm with Auto ML Natural Language in Google Cloud. By using the feature sentiment analysis, models can inspect a document an identify the dominant emotional opinion it contains and label he user's attitude as positive, negative or neutral.

4. Language & Keyword

With the metric *language*, it is possible to retrieve transcripts in a specific language. Next, you can specify your query with a *keyword*. In Dialogflow language is monitored under the metric languageCode. You can retrieve the *languageCode* from the query result from which intent is detected, in a dectedIntentResponse. For example:

```
queryResult.languageCode
```

Specific keywords don’t have to be stored. You can retrieve these based on user *utterances* because every language metric has to be stored under a *session ID* and a user *utterance* in a data warehouse. This way, language can be connected to other metrics which fall under the same session ID. With the right SQL query, you can now retrieve all messages containing a specific keyword in a particular language. For example, if you look for the Dutch word “Fraude”, the SQL query is as follows:

```
SELECT * FROM `chat_msg_table` WHERE LANGUAGE = “nl-nl” AND USER_UTTERANCE
LIKE ‘%Fraude%’
```

5. Platform
Monitoring the metric *platform* lets you retrieve all transcripts of a conversation held on a specific platform, for example, Google Assistant. You might need to set up this metric yourself, based on the implementation you are using. Like Google Assistant, an implementation has its own configurations set up for each device surface. However, if you’re building your own web interface with an integrated chatbot, you will need to come up with a name for your platform yourself, for example:

```
const platform = ‘web’;
```

Again, each platform metric should be saved under a *session ID* in your data warehouse, connecting the platform on which conversations take place to all the other metrics under that same session ID. For example, if you choose ‘*web*’ as the name for your web interface, you can retrieve all messages sent through that interface with the following SQL query:

```
SELECT * FROM `chat_msg_table` WHERE PLATFORM = “web”
```

6. Intent Detection
There are a few essential metrics that fall under *intent detection information: detected intent name, confidence threshold* (level of confidence), *if it’s a fallback message or not* and *if it’s an end-of-interaction message or not*. These metrics can help determine whether bots are functioning well or not.

In a data warehouse, you need to store the data fields for these metrics yourself. In a bot-building tool like Dialogflow, these fields are set up automatically and can be found in the intent detection results of queries:

1. The detected intent name can be found in the queryResult.intent from a detectIntentResponse:

```
queryResult.intent.displayName
```

2. With the *Boolean value: isFallback*, you can determine if a message is a *fallback message*. The *Boolean value* can be found in the *queryResult*. intent from a *detectIntentResponse*:

```
queryResult.intent.isFallback
```

3. With the *Boolean value: endInteraction*, you can also determine if a message is an *end-of-interaction message*. The *Boolean value* can be found in the *queryResult.intent* from a *detectIntentResponse*:

```
queryResult.intent.endInteraction
```

4. The *confidence threshold* can be found in the *queryResult* from a *detectIntentResponse*:

```
queryResult.intentDetectionConfidence
```

2.3 Building bots is an ongoing process

Unless you’re building a chatbot or voicebot for logged-in users only, it’s impossible to connect session ID’s to users and find specific transcripts. So, to find particular transcripts of individual users, you need to combine the metrics mentioned above and use them to filter your data. For example, if you search your data warehouse for specific chat session that happened yesterday at 2 PM and you know the user was unhappy, was using the Dutch Google Assistant and was talking about a malicious bank transaction, odds are that you probably have enough information to find the specific transcript.

In conclusion, gaining insights based on conversation-related metrics can help improve the conversations your customers have with your chatbot or voicebot . Collecting data under the metrics mentioned above can give you the answers you need to do this. What are the most frequently asked questions? How do customers see your brand? Are they satisfied? It is this kind of information that can help you improve the virtual agent behind your bot. It is clear: building bots is an ongoing process.

03

MEASURING YOUR SESSION FLOW DESIGN

Eight chat session & funnel metrics to monitor

03

MEASURING YOUR SESSION FLOW DESIGN

Eight chat session & funnel metrics to monitor

Two other important categories of metrics are chat session & funnel metrics. A funnel is the overall set of steps users take to achieve the desired goal. For example, a user in a web store, before making a purchase (the desired goal), first views several products (step 1), chooses a product (step 2), adds it to his or her shopping cart (step 3) and eventually makes the purchase (step 4). Regarding chatbots or voicebots, we also speak of chat funnels, which consist of the steps users take in conversing with your bot.

This chapter discusses eight crucial chat session & funnel metrics to monitor for getting a clear idea of your session flow design, and the set of steps users take in engaging your voicebot or chatbot. This allows you to determine the points of improvement within your session flow design and make users move through the funnel more smoothly. We will also discuss a couple of chatbot analytics platforms, Dialogflow and Chatbase, to explain how to keep track of these metrics.

What is a session(flow)?

A *session* represents one conversation in which user interacts with your virtual agent. Sessions can either be complete or incomplete (when users decide to stop responding to your bot). Each conversation is logged and stored.

A *session flow* is a visualisation of the most common journeys users have interacting with a bot.

3.1 Which chat session and funnel metrics are important?

Chat session and funnel metrics visualise the conversational route users take in engaging your bot. Within your conversational analytics, the following eight chat session and funnel metrics are relevant to monitor:

1. Total Usage
This is the number of people that used your chatbot or voicebot

2. The number and percentage of correct intent matches
To ensure your bot is working effectively the number and percentage of correct intent matches are two important metrics to monitor. Imagine you are an insurance company, and you have a virtual assistant for a voice or a chatbot on your website. The main feature of this bot is to provide help for filling in a declaration form, but it can also answer other general questions about the insurance company. You might be interested to know the number of people that entered the “declaration session flow” compared to the total number of people that interacted with your virtual agent.

3, 4, 5. Completion rate, drop-off rate, drop-off place
Taking the same use case of the insurance company bot, you also want to know the percentage of customers that have completed the ‘claim’ session flow, in other words: the completion rate.
If you notice that many users are not completing the flow, it is wise to monitor the drop-off place, to see where they end the session, as well as the drop-off rate, to see how many users drop off compared to the total number users. Monitoring these metrics provides you with a lot of information about your bot’s performance, and more importantly: it gives you an indication of what part of your bot you should improve.

6. User retention
User Retention is the continued use of a product or feature by your customers.

7. Endpoint health
For example, when you integrate with a web service or database to retrieve information and the server returns an error or is very slow in returning the information, this might crash your virtual agent, or at least give a “not so nice” user experience.

8. Discovery
Google Assistant sometimes recommends specific Actions, but only if it sees a match between a query and an Action – even though users didn’t formulate the exact Action’s invocation name. So, if you’re using Google Assistant, it is interesting to find out how users came upon your Action and with which queries.

3.2 Monitoring chat funnel & Session metrics

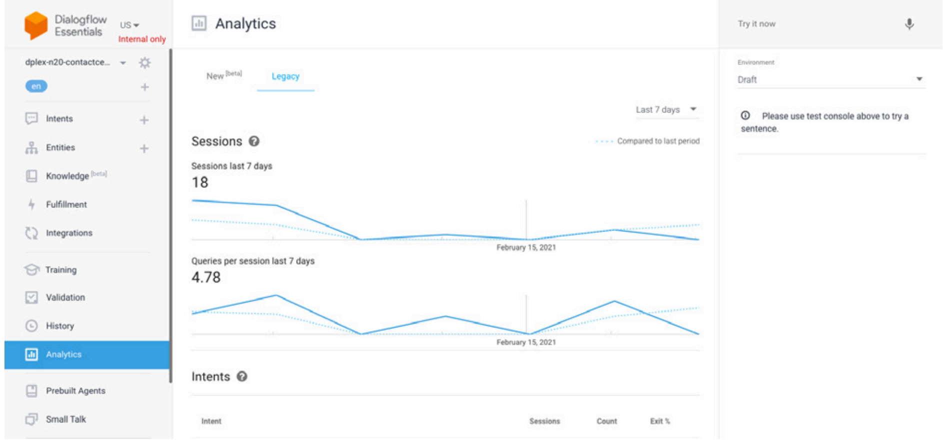
Navigating chatbot analytics platforms can be quite tricky. This paragraph explains how to monitor all of the chat funnel & session metrics mentioned above in analytics platforms Dialogflow and Chatbase.

3.3.1 Using Dialogflow

The *total usage* can be found by clicking on: ‘**Legacy > Explore**’. This page shows the total number of sessions within a period of your choosing (yesterday, last 7 days, or last 30 days).

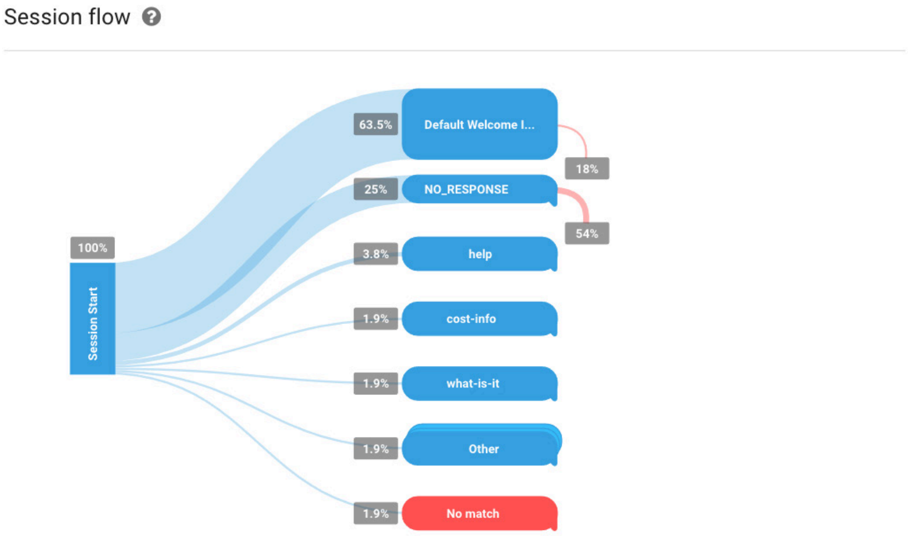
Scrolling down, you will see the session flow, depicting each intent. Click on each intent to reveal the follow-up intents. Hover your cursor over the intent names (blue boxes) to see the following metrics:

- The intent name
- The percentage of all users that were matched to the intent.
- The number of queries (messages) the intent was matched to.
- The drop-off rate (exit) following the intent match



Please use test console above to try a sentence.

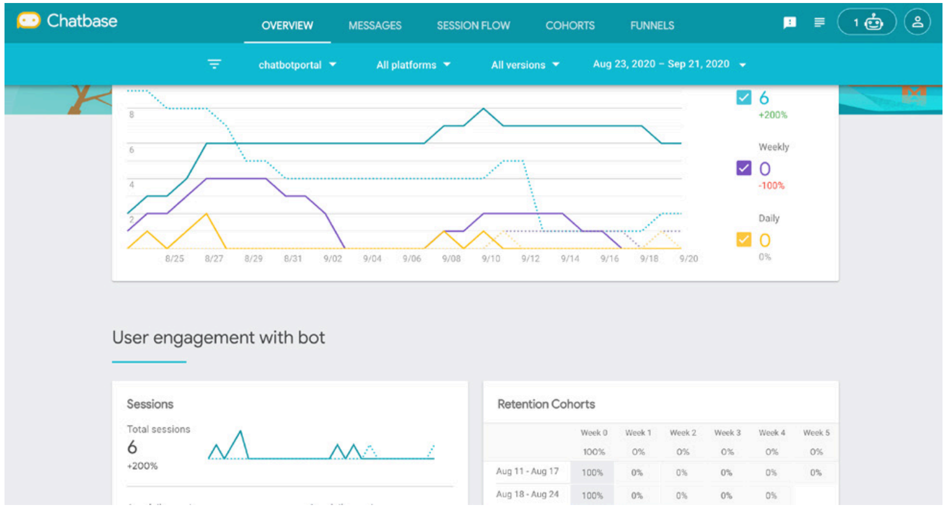
Explore page in Dialogflow



Session flow in Dialogflow

3.3.2 Using Chatbase

Alternatively, you can monitor these metrics in Chatbase. Chatbase gives you a more detailed overview and offers more options to filter your data by time (extra filters: ‘today’ and ‘quarter to date’), including a **Retention Cohorts** block, which shows user retention over time.



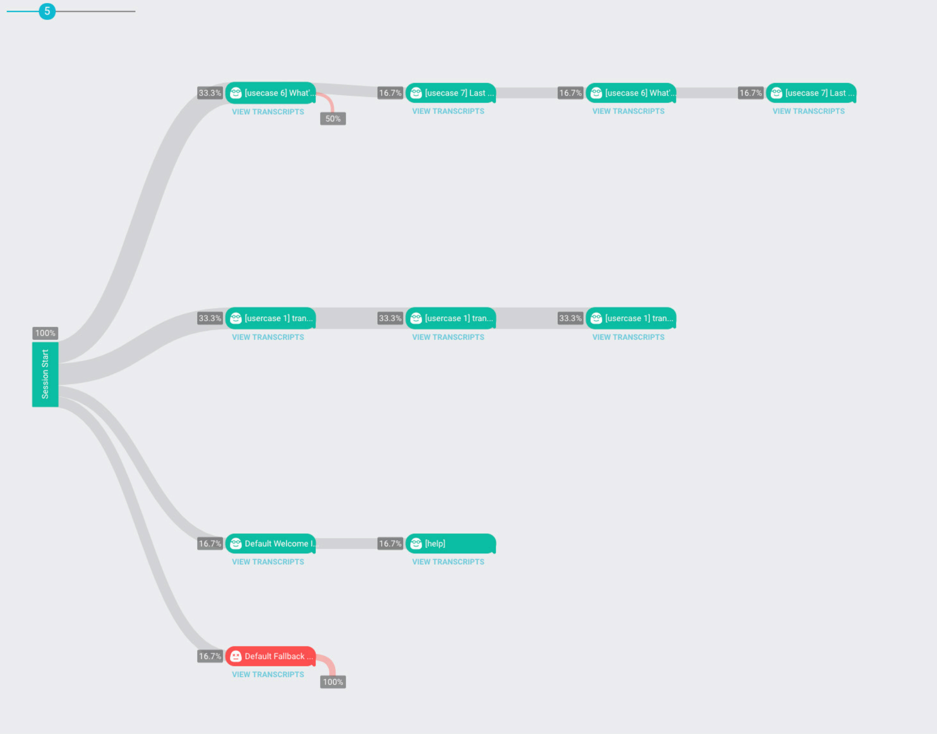
Retention Cohorts

	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5
	100%	0%	0%	0%	0%	0%
Aug 11 - Aug 17	100%	0%	0%	0%	0%	0%
Aug 18 - Aug 24	100%	0%	0%	0%	0%	

Overview page in Chatbase

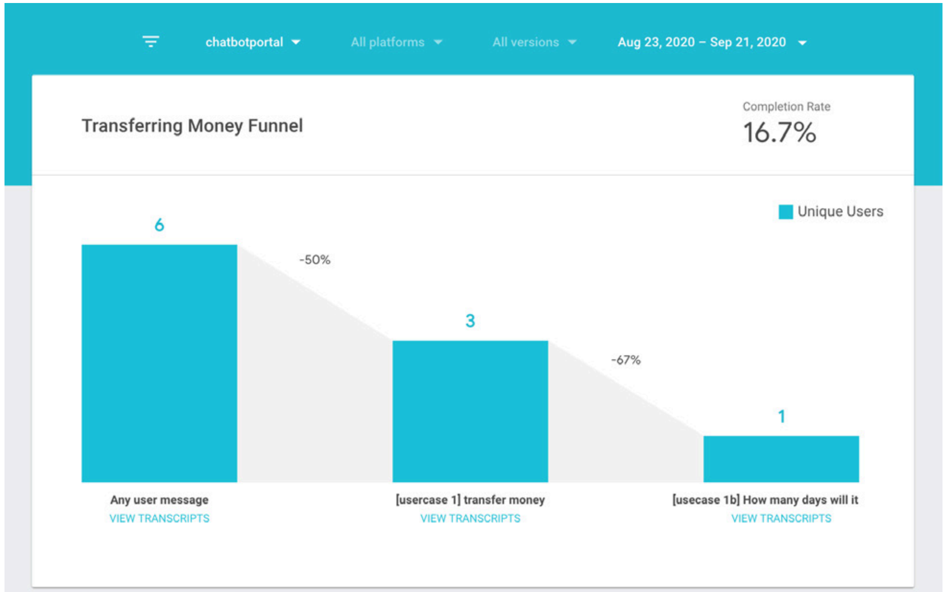
To view the session flow, click on the ‘Session Flow tab’. Here you will find the following metrics:

- The intent name
- The percentage of all users that were matched to the intent.
- The number of requests the intent was matched to.
- The drop-off rate while being on this intent


5
100%
Session Start
33.3% [usecase 6] What'...
VIEW TRANSCRIPTS
50%
16.7% [usecase 7] Last ...
VIEW TRANSCRIPTS
16.7% [usecase 6] What'...
VIEW TRANSCRIPTS
16.7% [usecase 7] Last ...
VIEW TRANSCRIPTS
33.3% [usercase 1] tran...
VIEW TRANSCRIPTS
33.3% [usercase 1] tran...
VIEW TRANSCRIPTS
33.3% [usercase 1] tran...
VIEW TRANSCRIPTS
16.7% Default Welcome I...
VIEW TRANSCRIPTS
16.7% [help]
VIEW TRANSCRIPTS
16.7% Default Fallback ...
VIEW TRANSCRIPTS
100%

Session flow in Chatbase

Another feature of Chatbase is the funnel reporting tool, which offers a way to measure customer success by creating custom workflows for up to six follow-up intents. Click on the 'Funnels' tab and start creating new funnels to record and assign intents to turn-taking steps. Once you have created a funnel, you get a clear image of the conversation route users take, and whether or not they complete the whole funnel (*completion rate and drop-off rate*).

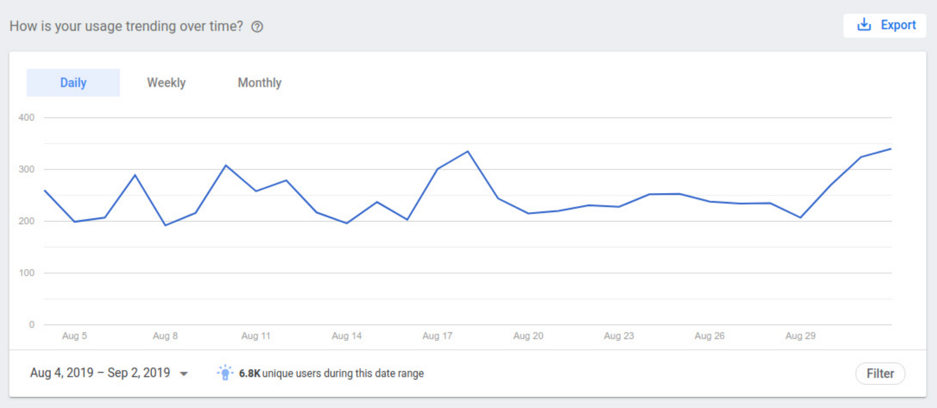


Creating funnels in Chatbase

3.4 Monitoring Chat Session & Funnel metrics from Google actions

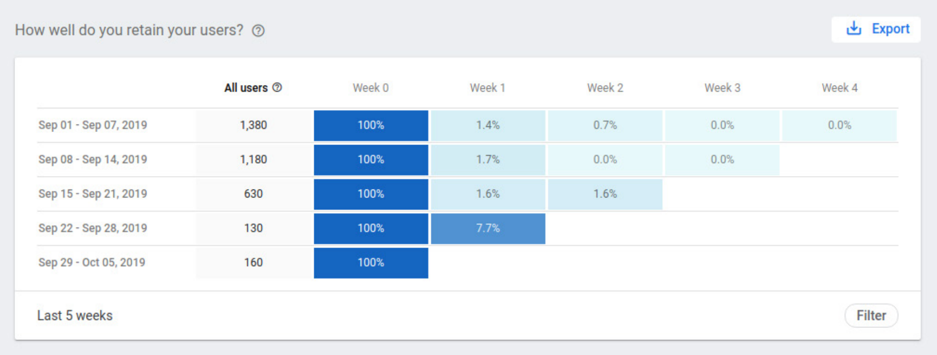
When building a voicebot for Google Assistant, you can configure it to monitor selected metrics in Google Console. The benefit of this is that you can export these metrics to BigQuery to store data for a longer period.

By clicking on '**Analytics > Usage**', you arrive on the Usage page, displaying three graphs depicting your Action's usage data over time.

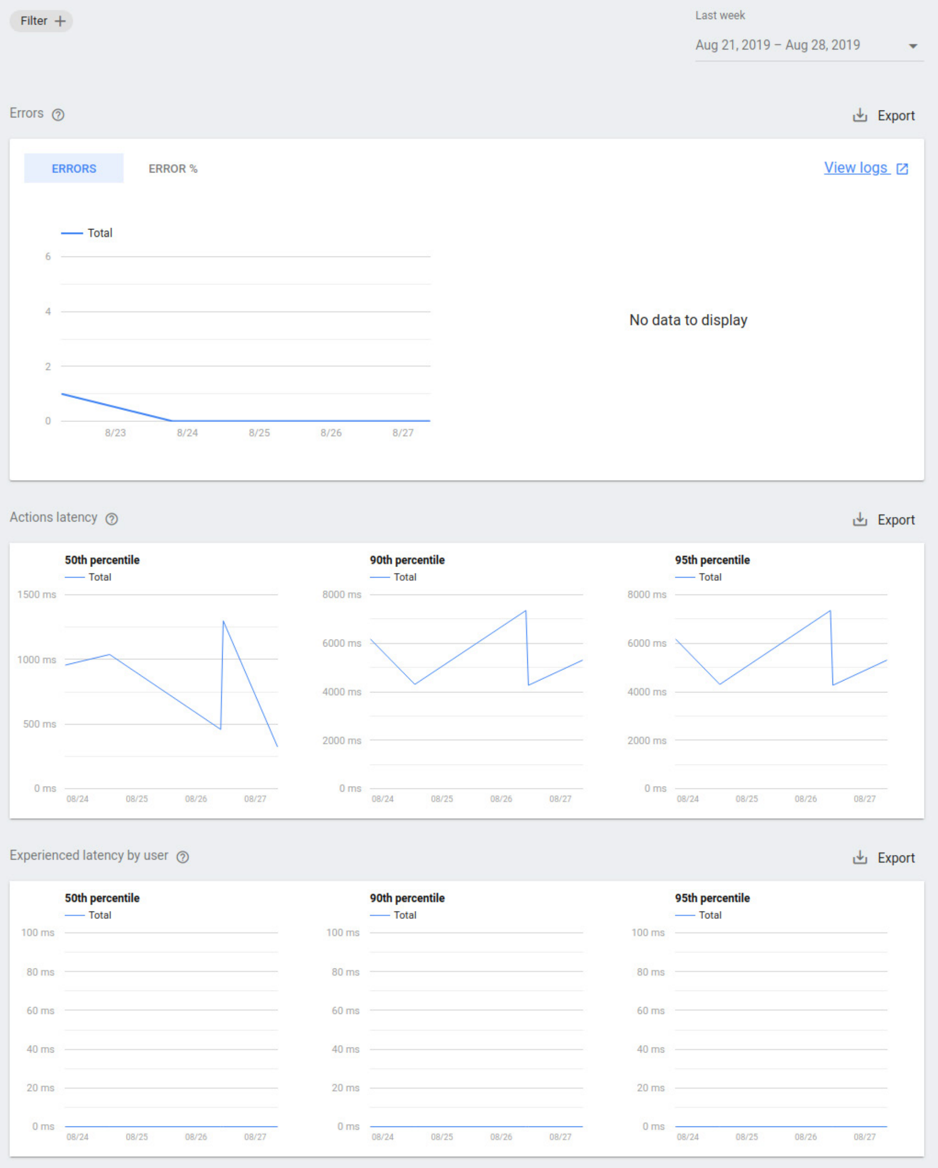

How is your usage trending over time?
Export
Daily
Weekly
Monthly
400
300
200
100
0
Aug 5
Aug 8
Aug 11
Aug 14
Aug 17
Aug 20
Aug 23
Aug 26
Aug 29
Aug 4, 2019 – Sep 2, 2019
6.8K unique users during this date range
Filter

Usage page in Google Console

It also shows your Action's **user retention**:



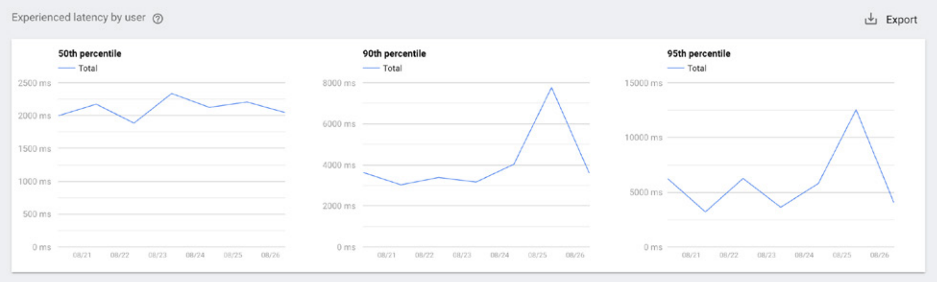
	All users	Week 0	Week 1	Week 2	Week 3	Week 4
Sep 01 - Sep 07, 2019	1,380	100%	1.4%	0.7%	0.0%	0.0%
Sep 08 - Sep 14, 2019	1,180	100%	1.7%	0.0%	0.0%	
Sep 15 - Sep 21, 2019	630	100%	1.6%	1.6%		
Sep 22 - Sep 28, 2019	130	100%	7.7%			
Sep 29 - Oct 05, 2019	160	100%				



Health information page in Google Console

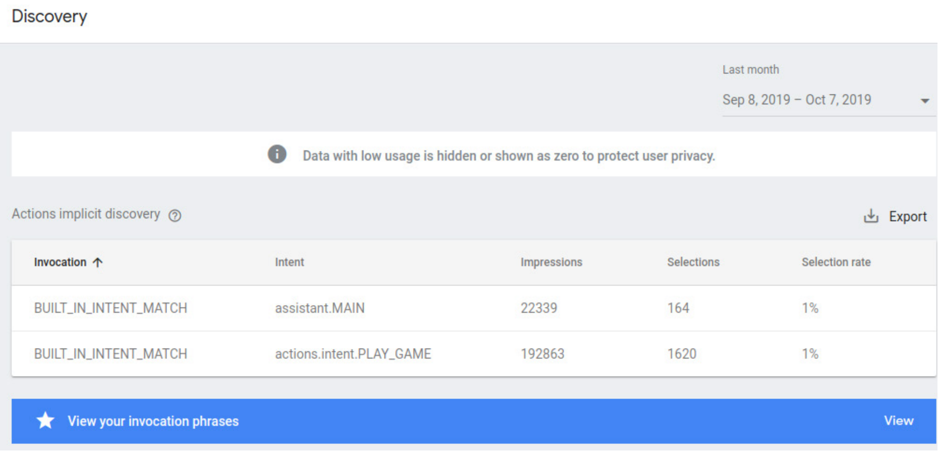
Another benefit of using Google Actions is that it shows you *Health information* reports categorised under the following metrics:

- **The number of *errors*** an Action's cloud endpoint returned on a given day. If you have a lot of errors, you may want to look at your logs to identify what is causing your endpoint to crash or behave unexpectedly.
- **Action latency**: The latency of your Action's endpoint. If latency is very high or spikes regularly, your users may be experiencing delays while interacting with your Action.
- **Experienced latency by user**: The latency felt by a user on each request to your Action. This metric illustrates what users experience when interacting with your Action.



By clicking on ‘**Analytics > Discovery**’, you arrive on the Discovery page. This page displays a table containing the phrases that prompted Google to recommend your Action. The metrics shown in this table are:

- **Invocation**: The user query that prompted Google to recommend your Action. Aside from the actual user queries, this list includes the following values:
  - BUILT_IN_INTENT: This listing indicates your Action was invoked through a built-in intent.
  - AUTO_MATCHED_BY_GOOGLE: This listing indicates when implicit invocation was used.
  - ACTION_LINK: This listing indicates when your Action was invoked through an Action link.
- **Intent**: The intent to which the user’s query was matched.
- **Impression**: The number of times this phrase prompted Google recommending your Action.
- **Selection**: The number of times a user invoked your Action after Google recommended it. This selection number of a phrase cannot exceed the number of impressions.
- **Selection rate**: The percentage of impressions that prompted a selection. A low selection rate indicates that many users choose to use other Actions for this particular query, whereas a high rate suggests that your Action is popular for this query.



Discovery

Last month

Sep 8, 2019 – Oct 7, 2019

Data with low usage is hidden or shown as zero to protect user privacy.

Actions implicit discovery

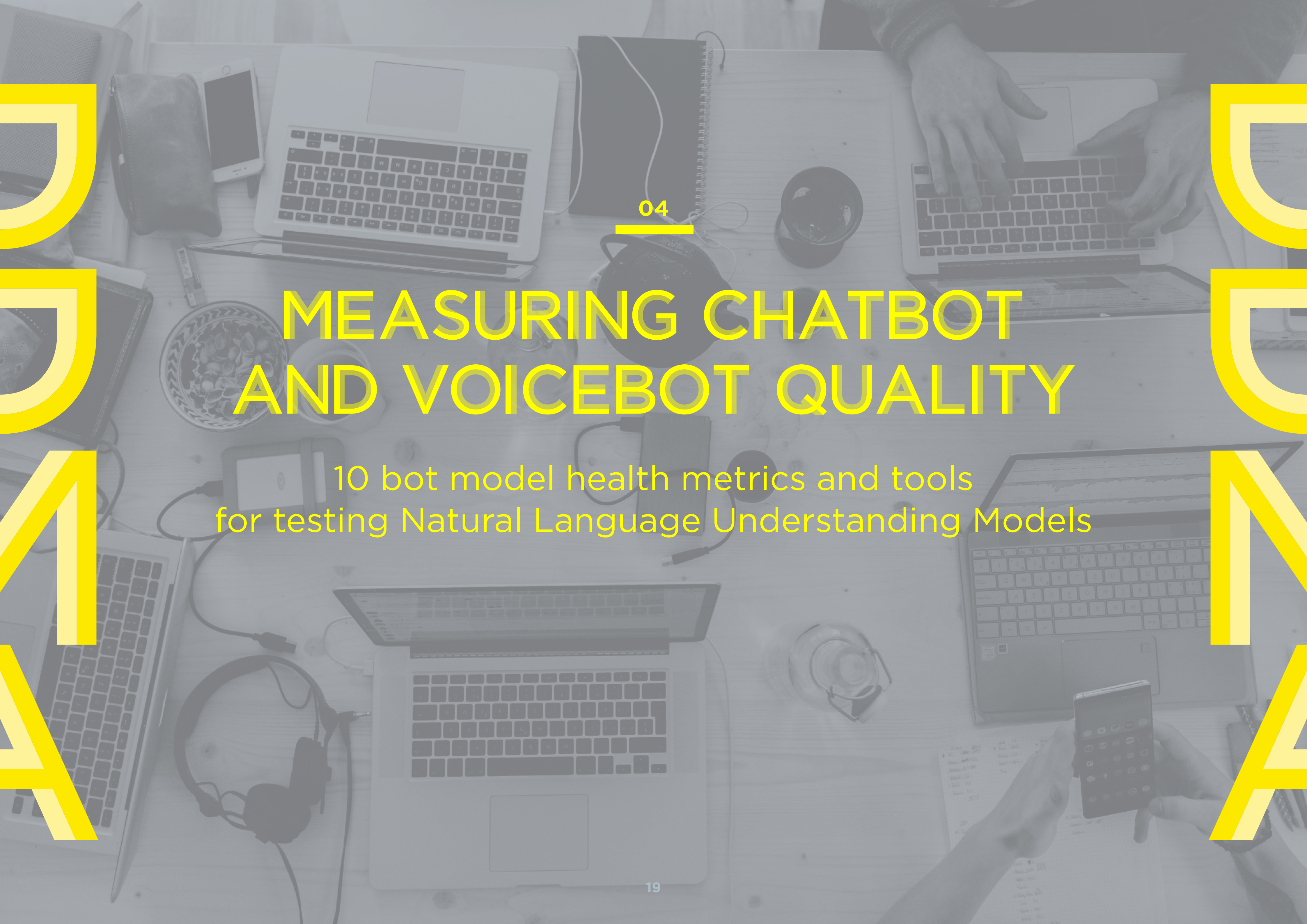
Export

Invocation ↑	Intent	Impressions	Selections	Selection rate
BUILT_IN_INTENT_MATCH	assistant.MAIN	22339	164	1%
BUILT_IN_INTENT_MATCH	actions.intent.PLAY_GAME	192863	1620	1%

View your invocation phrases

View

Discovery page



04

MEASURING CHATBOT AND VOICEBOT QUALITY

10 bot model health metrics and tools
for testing Natural Language Understanding Models

MEASURING CHATBOT AND VOICEBOT QUALITY:
10 bot model health metrics and tools for testing Natural Language Understanding Models

While conversation-related metrics and chat session & funnel metrics revolve around collecting data on user experience, bot model metrics can help improve the actual bot-building process and the quality of the underlying Natural Language Models. They can help you to make sure your bot's intent matching and classification are done correctly. This chapter goes into greater detail about the ten *bot model health metrics*, crucial to mapping the quality of your chatbot or voicebot .

4.1 What is Natural Language Understanding?

A bot uses Natural Language Understanding (NLU) to match a user utterance to a specific intent. It is a form of machine learning that enables chatbot or voicebots to determine automatically what users mean, regardless of how they express themselves. In doing so, the bot is able to match user expressions to specific intents. However, to create a properly-functioning NLU model, it needs to be trained with phrase examples, i.e. test cases used to make future intent matched. This is where the monitoring of bot model health metrics comes in to play.

4.2 Which bot model health metrics are important?

To determine whether intent matching is being done correctly, there are ten bot model health metrics you must monitor if you want to determine whether your bot correctly matches user utterances to specific intents:

1. True Positive - A correctly matched intent
A **true positive** outcome is when a bot correctly matches a user utterance to the right (positive) intent. For example:

> A bot was trained to recognise an intent called *salary intent* with the phrase *"Did my salary come in?"*. If the bot still matches salary intent to a user utterance like *"Did I receive my salary?"*, we speak of a *true positive*. The intent was matched correctly.

You can write a test case to check if the *user utterance* matched the *expected intent name*. You will also need to know the actual detected intent name. By storing these fields together with the *result* – true positive or not – in a data warehouse, you can rerun the scenario each time you make changes to your bot.

2. False positive - A misunderstood request
A **false positive** outcome is when a bot matches a user expression to the wrong intent. The expression should either be matched to a different or a fallback intent (in case a matching intent doesn't exist). Misunderstood request errors occur when a bot can't determine the correct user intent. For example:

> A bot was trained to recognise an intent called *Block credit card* with the phrase *"I want to block my credit card."* It is also trained to recognise an intent called Renew credit card with the phrase: *"I want to renew my credit card."* So, a user utterance like *"My account is blocked, can I get a new credit card?"* is expected to be matched to the *renew credit card* intent. It may be possible though that this phrase is matched to the wrong *Block credit card* intent. In such a case, we speak of a *false positive*.

You can write a test case to check if the *user utterance* did not match the *expected intent name*. You will also need to know the *actual detected intent name*. By storing these fields together with the *result* – false positive or not – in a data warehouse, you can rerun the scenario each time you make changes to your bot.

3. A true negative – An unsupported request
A **true negative** is an outcome where the bot correctly matches a user utterance to a fallback intent. This happens when users ask questions to which a bot doesn't have the answers, resulting in a fallback message. For example:

> Every time a bot cannot answer a question, it matches the user utterance to an intent called *Global fallback*. If a user asks: *"Can I buy casino tokens?"* to a banking chatbot, it is very likely the user utterance is matched to the Global fallback intent. In this case, we speak of a *true negative*. In this example, the bot behaves exactly the way it was designed.

You can write a test case to check if the *user utterance* matched the *expected intent name*, which should be the *Global Fallback Intent*. You will also need to know the *actual detected intent name*. By storing these fields together with the *result* – true positive or not – in a data warehouse, you can rerun the scenario each time you make changes to your bot.

4. False Negative - A missed request
A **false negative** is an outcome where bots unnecessarily match a user utterance to a fallback intent, meaning the right intent to answer a question does exist, but the bot failed to detect it. Missed request errors occur when a bot has the right intents but fails to recognise them because of alternative phrasing or terminology. For example:

> A bot was trained to recognise an intent called *Block credit card* with the phrase *"I want to block my credit card."* If a user utterance like *"Stop the credit card right now"* is matched to a *fallback intent*, we speak of a false negative. The intent exists, but the bot was unable to make the correct match.

You can write a test case to check if the *user utterance* did not match the *expected intent name*, but instead matched a *fallback intent*. You will also need to know the *actual detected intent name*. By storing these fields together with the *result* – false negative or not – in a data warehouse, you can rerun the scenario each time you make changes to your bot.

5. Accuracy
Accuracy is the ratio of correctly predicted observations to the total number of observations. In other words: the ratio of all the correct matched intents to the total number of matched intents. You can determine accuracy as follows:

```
total correct = total true positives + total true negatives
total incorrect = total false positives + total false negatives.
accuracy = correct / correct + incorrect
```

You can write a test case and store the results in your data warehouse so you can rerun the scenario each time you make changes to you agents. If you want to do this, you need to store the accuracy and the total number of true positives, *true negatives, false positives* and false *negatives* in your data warehouse.

6. Precision
Precision refers to the ratio of correctly matched intents to the total number of matched intents. If false positives occur frequently, precision will be low.
In other words: the higher the *precision*, the lower the false positive rate.
You can determine *precision* as follows:

```
precision = total true positives / (total true positives + total false
positives)
```

You can write a test case and store it in a data warehouse, enabling you to rerun the scenario each time you make changes to your agent. To do this, you need to request the total number of *true positives* and *false positives* from your data warehouse. You then need to store the *precision* so you can retrieve it later on in your reports.

7. Recall & Fallout

The *recall (True Positive Rate)* is a sensitivity ratio. To determine whether intents are too narrowly defined, causing requests to be missed and resulting in too many false negatives. A recall of higher than 0.5 is considered good. Recall can be determined as follows:

```
recall = total true positives / (total true positives + total false negatives)
```

The *Fallout (False Positive Rate)* is a false alarm ratio: the number of user utterances matched to the wrong intents to the total number of false positives and true negatives combined. The Fallout can be determined as follows':

```
fallout = total false positives / (total false positives + total true
negatives
```

You can write a test case and store it in a data warehouse so you can rerun the scenario each time when you make changes to your agent. You would need to request the total number of *true positives, false positives, false negatives* and *true negatives* from your data warehouse. You also need to store the **recall** and **fallout rates** so you can retrieve them later on in your reports.

8. F1 Score

The **F1 score** is the weighted average of precision and recall. This score therefore takes both false positives and false negatives into account. Intuitively, it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have a similar cost. If the costs of false positives and false negatives are very different, it's better to look at both Precision and Recall.

```
F1 Score = 2 * (Recall * Precision) / (Recall + Precision)
```

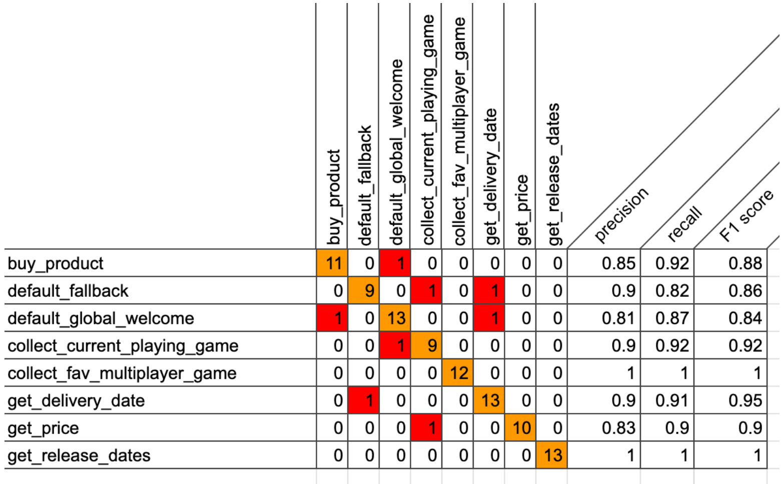
You can write a test case and store it in a data warehouse, so you can rerun the scenario each time you make changes to your agent. You would need to request the *total recall* and *precision* from your data warehouse. You also need store the *F1 score* so you can retrieve it later on in your reports.

9. Confusion Matrix

In the field of machine learning, a *confusion matrix*, also known as an error matrix, is a specific table layout used to visualise the performance of an algorithm – in this case, the underlying machine-learning model for chatbots or voicebots. Each row in the matrix represents the expected intents while each column stands for the actual matched intent by the Dialogflow API.

		Actual Values	
		Positive (1)	**Negative (0)**
Predicted Values	**Positive (1)**	TP	FP
	Negative (0)	FN	TN

The name derives from the fact that a confusion table makes it easy to determine whether bots are getting two classes confused with each other, commonly mislabelling one as the other. When you have lots of test cases, you could render all the scores in one big matrix.



	buy_product	default_fallback	default_global_welcome	collect_current_playing_game	collect_fav_multiplayer_game	get_delivery_date	get_price	get_release_dates	precision	recall	F1 score
buy_product	11	0	1	0	0	0	0	0	0.85	0.92	0.88
default_fallback	0	9	0	1	0	1	0	0	0.9	0.82	0.86
default_global_welcome	1	0	13	0	0	1	0	0	0.81	0.87	0.84
collect_current_playing_game	0	0	1	9	0	0	0	0	0.9	0.92	0.92
collect_fav_multiplayer_game	0	0	0	0	12	0	0	0	1	1	1
get_delivery_date	0	1	0	0	0	13	0	0	0.9	0.91	0.95
get_price	0	0	0	1	0	0	10	0	0.83	0.9	0.9
get_release_dates	0	0	0	0	0	0	0	13	1	1	1

Example of a confusion matrix

As you can see, each orange square contains the sum of all the *true positives* of a certain intent, resulting in an orange diagonal (orange) line of squares from the top to bottom. The red squares contain the sum of all the false positives. For example, this bot's confusion matrix tells us that a user utterance was matched to an intent called *default_fallback*, whereas it should have been matched to the intent *get_delivery_date*.

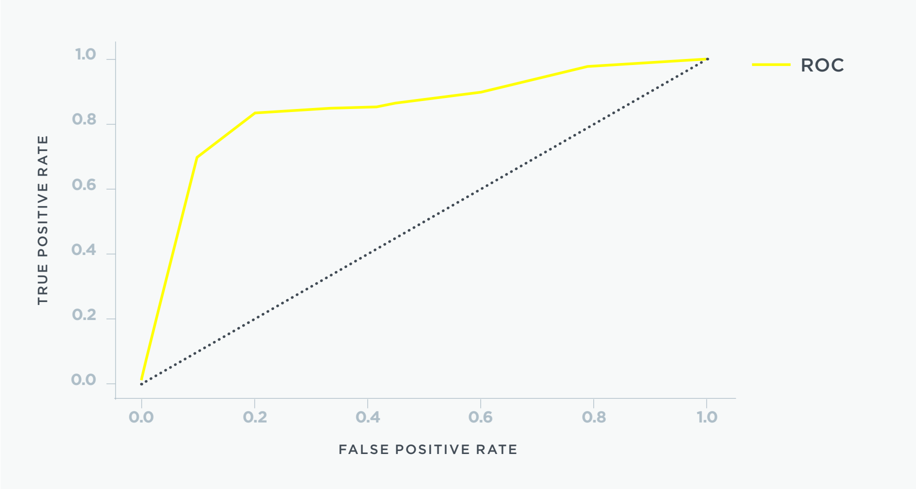
10. Receiver Operating Characteristic (ROC) Curve
A *ROC (Receiver Operating Characteristic) Curve* is a graphical representation of how well a model can distinguish the given intents in terms of detected probability. Using this graph, you can set up confidence thresholds for your virtual agent by defining the probability score required for an intent match.

Let's say, for instance, that we have set the confidence threshold in Dialogflow to 0.80. This threshold divides user utterances into two options: intent match or no intent match. A probability higher than or equal to the threshold of 0.80 results in an intent match. A probability above 0.80 results in a match with a fallback intent.

The ROC Curve plots two parameters:

- **True Positive Rate (recall / sensitivity ratio)**
- **False Positive Rate (fallout / false alarm ratio)**

If the ROC curve is similar to the blue diagonal line, it means that virtual agents correctly match half of the expected intents. In general, the goal is to maximise sensitivity and false alarm. In other words: the steeper the ROC (yellow) curve, the better.



Receiver Operating Characteristic (ROC) Curve

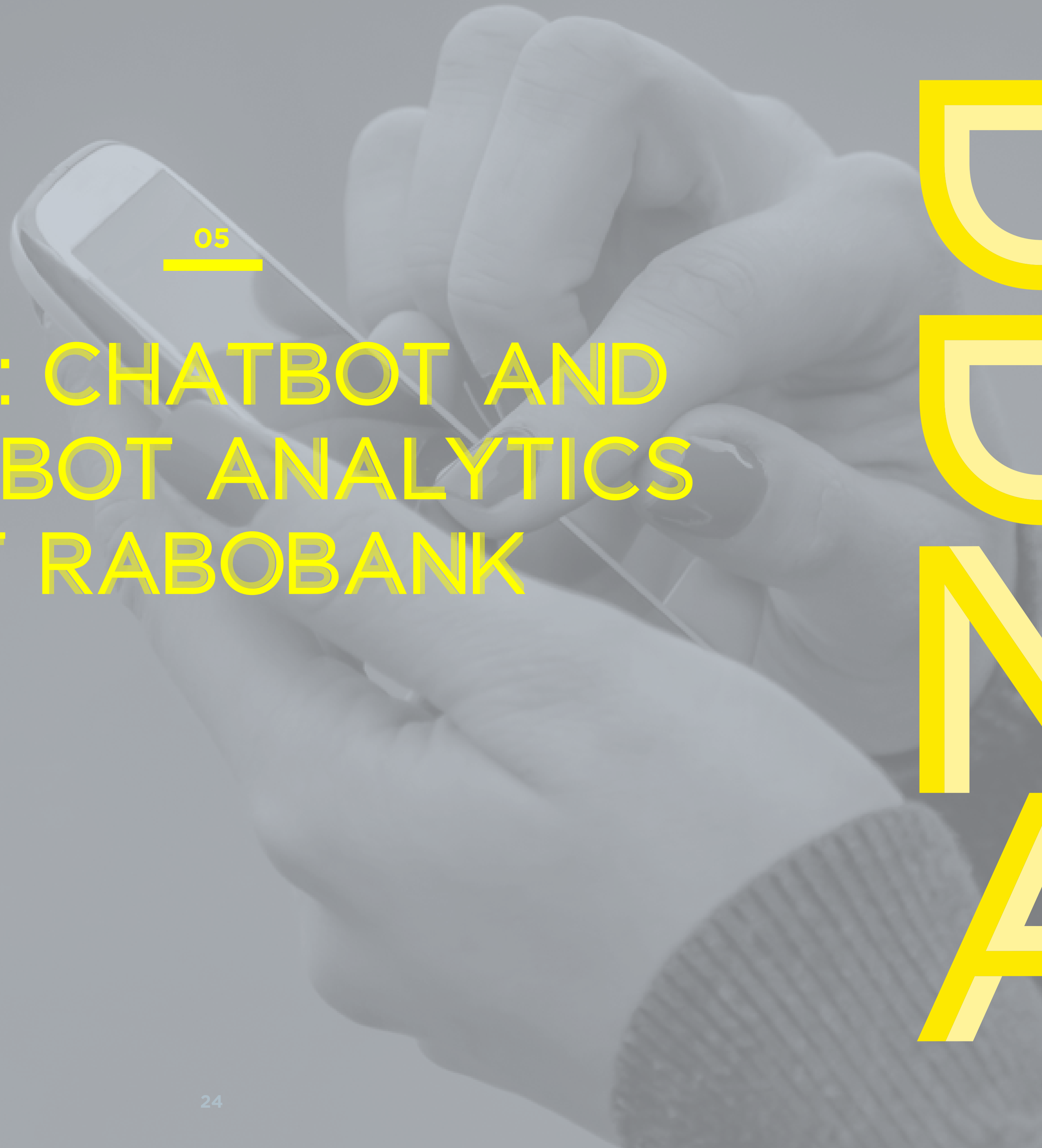
4.3 Make sure your data sets aren't biased

Hopefully, this will give you an idea of the benefits that monitoring chatbot model health metrics offer. They are great metrics to test if your underlying machine-learning model works the way you want it to work. Once you have stored all these metrics in a data warehouse, you can easily create dashboards to retrieve the insights you need. For instance, developers can create unit tests with user utterances based on validation data, and test them against a bot builder API (in Dialogflow this will be the *detectIntent() method*). The detected intent and the confidence score can be evaluated with your validation dataset.

Typically, when working with test data, UX designers or content writers create a validation data set to train a Dialogflow agent model by entering it as user phrases. However, you have to be careful not to create test data sets that are biased. You can prevent this by using logs from chats, contact centres, and virtual assistants, and separate them from the intent training phrases that were used to train the model. If user data contains sensitive personal data, you should anonymise or mask it.

05

CASE: CHATBOT AND VOICEBOT ANALYTICS AT RABOBANK



05 CASE: CHATBOT AND VOICEBOT ANALYTICS AT RABOBANK

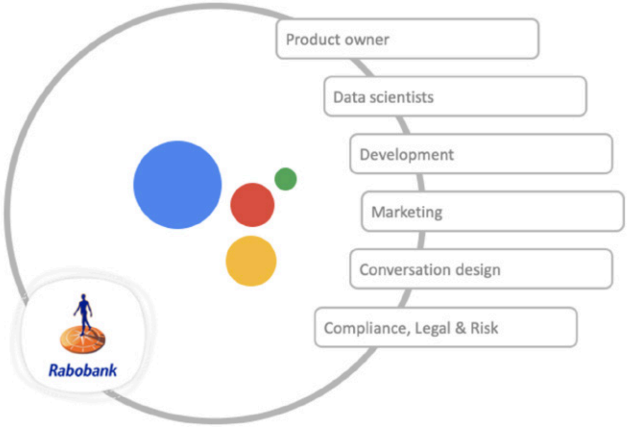
Several Dutch organisations deploy conversational platforms as a great way to communicate with their customers, and they use analytics to allow the ongoing optimisation of their voice and chat channels. Rabobank is such an organisation. Conversational design initiatives at Rabobank have led to two conversational channels in the past three years: the virtual assistant, in the form of a chatbot, and the Rabo Assistant, a Google Assistant voicebot . In this chapter, we show which data sources and analysis forms Rabobank uses so it can keep on refining both of these channels.

5.1 The Rabobank Virtual Assistant and the Rabo Assistant

The virtual assistant: a conversational chatbot
Three years after the start of the conversational platform project, Rabobank's virtual assistant has become a fully-fledged chatbot. The chatbot is the first moment of contact for all customers using the chat tool on either the Rabobank website or in the banking app. If the questions become too complex, then the bot puts the customer through to an agent. The creation of this bot required not only the expertise of existing teams, it also involved the setting up of a number of new teams (see table x for all teams involved).

Rabo Assistant: a conversational voicebot
As a launching partner of the Dutch Google Assistant, Rabobank has now launched one of the first Dutch voice actions. This made it the first Dutch organisation with which customers could communicate using voice. Among other things, customers can now check their balance, view a list of their latest transactions, send a payment request or carry out a simple mortgage calculation.

TEAMS INVOLVED	
Service Content & Dialogues team	Responsible for the creation of service content for web, app and the virtual assistant.
Virtual Assistant Development Team	Responsible for the technology and platform behind the virtual assistant
Search & Voice Development Team	Responsible for the technology behind search options and the Rabo Assistant
Rabobank Customer Service	Responsible for processing live chat and telephony
Data & Insights	Responsible for everything connected with data and customer insight(s). This can vary from setting up a KPI dashboard to unlocking application data in a data lake. However, the vast majority of data analysis is done by the Conversational Team itself.



5.2 Initial topics and temporary answers

Just over a year before the launch, a start was made on the conversational content for the virtual assistant. The first step was to develop five topics, based on keyword research, live chat and traffic analysis of the most visited pages on Rabobank.nl: BIC / Swift / IBAN, credit card, debit card, Rabo Scanner and Rabo Payment Request. When it went live, it was not yet possible for the chatbot to answer questions on subjects that had not yet been developed. To prevent the virtual assistant from responding in many cases with, *"I don't understand what you are saying"* or *"Could you rephrase that?"*, general temporary answers were formulated for incomplete topics. For example: *"I understand that you have a question about mortgages. I will be able to tell you all about them shortly, but for now, it would better if a member of staff were to help you. Shall I put you through to a member of staff so you can chat with them?"* Now - following the launch of the virtual assistant - Rabobank uses customer queries to gain insight into their needs and determine which topics need to be developed so they are more relevant.

5.3 Which data sources does Rabobank use to refine content in the virtual assistant and Rabo App?

Besides customer queries, Rabobank uses other processes, reports and tooling to determine which new content is required and which content they have to refine. Of all the available data, these are the processes/sources that Rabobank uses the most:

1. Conversation Review
Each day, two people review chatbot conversations from the previous day. The conversations are then linked to certain intents (see figure x). It's a time-consuming job, but a valuable one because:

- If there are two of you, you keep each other alert to recognising the actual question behind consumer queries.
- You can recognise early on the problems that customers are struggling with
- You can build up a decent image of the content customers need

As more conversations are conducted through the virtual assistant than through the Rabo Assistant, the conversation review takes place every day or every fortnight respectively.

What is intent?

Within a chatbot, 'intent' refers to the purpose the customer has in mind when asking a question or making a comment. To understand a customer's intent, you first have to train a chatbot using examples. Rabobank does this by creating dialogues that deal with specific intents. The more examples of a particular intent a chatbot has seen, the easier it will be to recognise it

2. Chat coaching
Agents used to read ongoing chat conversations that customers were having with Rabobank's virtual assistant. Whenever the chatbot was unable to figure out a question because it was unclearly worded, an agent would intervene and formulate an answer that responded better to what the customer had asked. This allowed Rabobank to provide a better customer experience while simultaneously training the chatbot's voice recognition skills. Rabobank has since stopped doing this. Instead, an advisory panel of former coaches has now been set up to review all content.

3. Customer service
The lines of communication between the team responsible for conversational content and customer service are kept short. The Rabobank Customer Service provides weekly reports on the reasons for customer calls. This is another way of gaining insights to aid building or refining content. High call volume topics are picked up quickly. For example, at the beginning of the coronavirus crisis, there were lots of questions about holiday bookings.

4. App Store ratings
Expressions of dissatisfaction with Rabobank's services have proved very useful. For example, App Store reviews are very valuable to Rabobank. One-star reviews in particular contain useful information that multiple teams can examine to gain greater insight into customer needs. Reviews often only need a few words to reveal what customers are struggling with. They are stored anonymously at Rabobank and displayed in a business intelligence dashboard. The dashboard even lets us view customer feedback on each individual feature. This is very valuable for development teams, as well as being a source of inspiration for new conversations. As such, there are various APIs that we use to capture the data from these reviews for our own use.

5. Feedback through Usabilla

Everywhere on the website, in the app and on the online banking platform, Rabobank asks customers to leave feedback through Usabilla. Several teams then use this feedback to refine content and functionalities, and improve the user experience.

6. Google Analytics / Datastudio

We use Google Analytics so we are better able to measure visitor behaviour, chatbot use and voice activity in our website. A number of handy dashboards have been built into Google Data Studio that provide us with useful insights into the performance of these channels.

5.4 Chat and voicebot analytics: what does Rabobank report on?

At Rabobank, key stakeholders, the customer service centre and developers receive monthly reports on chat and voicebot analytics. The latest information can also be viewed on a KPI dashboard. The metrics included in these reports differ for each channel (see tables on page 27-28). The following analysis tools are used for the reports: Google Analytics, Google Data Studio, Google Action Console, Nuance, Dialogflow and Microsoft Power BI.

ANALYTICS IN RABOBANK'S VIRTUAL ASSISTANT	
Number of chats	Total number of chats started on Rabobank.nl (anonymous website) and the banking environment.
Feedback score	At the end of the chat, users of the virtual assistant are given the opportunity to rate the experience and leave feedback. The scores are then used to measure the virtual assistant's performance. It also examines the scores given to chats that are transferred to an agent.
Accuracy test	This is an automated test in which the virtual assistant uses a representative set of training sentences and "forgets" which intent they belong to. The sentences are then run through the bot's grammar (Natural Learning Engine) as if they had been asked for the first time. After the test, checks are carried out on how many of these sentences have been correctly understood and matched to the right dialogue. The metric that Rabobank reports is the percentage of questions that are directly linked to the right intent; i.e. the correct intent rate.
Helpfulness	Each time a customer gives a final answer to a question asked by the virtual assistant, this is followed by an automated evaluation question (no more than one question per call): *"Was this answer helpful? Answer yes or no?"* The *"No"* answers are not always useful as feedback because rather than rating the correctness of an answer, customers often rates the eventual result of the answer. For example, if a customer wants to order tickets for a sold-out show, even though the bot might have answered the question correctly, the customer may still give it a poor rating. The metrics reported by Rabobank consist of: • The "Yes" answers as a percentage of the total number of answers (Yesses + Nos). • The number of users who answered the evaluation question and the percentage they represent (double with next bullet?) • The users who answered the evaluation question as a percentage of the total number of users asked.
Most common topics	Rabobank keeps track of which topics get asked about the most. This is to gain insight into which topic needs to be developed next in dialogues and separate intents.

ANALYTICS IN THE RABO ASSISTANT (ACTION IN GOOGLE ASSISTANT)	
Number of users	Number of active users on the Rabo Assistant.
Invoked intents	The invoked intents started by users
Device	Device on which the Rabo Assistant is being used: telephone, smart display or speaker.
Language	Language setting on the device used to access the Rabo Assistant.
PSD2 Consent	The number of active consents and their growth per month* **What is PSD2 consent?* PSD2 consent must be given so Google Assistant is granted access to a user's financial information. This consent grants Google 90 days of access to a limited amount of financial data (balance & transaction information). To extend this access, consent must be renewed every 90 days. Users may withdraw this consent at any time.
Number of interactions per call	The number of interactions a customer has during a call with the Rabo Assistant.
Assessment of Rabo Assistant	Customer rating of the Rabo Assistant on a scale of 1 to 5.

5.5 Next step: data storage in a data lake

The need for an enterprise data lake was emphasised earlier on in this white paper (see chapter 1). Rabobank has had an Enterprise Data Lake at its disposal for some time. The bank is currently busy using it to store all data from the virtual assistant. By doing so, Rabobank aims to be able to follow the customer across multiple channels and gain insight into the part each channel plays in the customer journey.

ABOUT DDMA

DDMA is the largest association for data-driven marketing, sales and service. We are a network of advertisers, non-profits, publishers, agencies, and tech suppliers that use data in an innovative and responsible way to interact with consumers. With knowledge and advice, we help our members to work in a data-driven and customer-focused way, to develop a vision on data use and to deal with legal changes. We also give our members a voice in The Hague and Brussels and we professionalise the sector by developing self-regulation.

QUESTIONS?

Do you have questions about this whitepaper? Contact us via info@ddma.nl or 020 4528413. All information about this project can be found at ddma.nl/ca.

WHAT IS YOUR OPINION?

We are very curious to hear what you think of this whitepaper. Help us by answering a few questions in this short questionnaire.