

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Sistemas Informáticos Avanzados

Sistemas distribuidos y web

Tesis fin de Máster

Sheetchat - Generación de chatbots a partir de hojas de cálculo.

Autor

Haritz Medina

informatika
fakultatea



facultad de
informática

2016

Resumen

Abstract of the project at the beginning of the document.

Índice de capítulos

Resumen	I
Índice de capítulos	III
Índice de figuras	VII
Indice de tablas	IX
1. Introducción	1
2. Introducción a los chatbots	3
2.1. Definición y contexto de uso	3
2.2. Plataformas para desarrollo de agentes conversacionales	5
3. Análisis y diseño de la solución	9
3.1. Requisitos	10
3.2. Modelo de características	10
3.2.1. Sheet	10
3.2.2. Chat	10
3.3. Modelo de datos	10
3.3.1. Metamodelo	10
3.3.2. Sintaxis concreta	10

4. Implementación	11
4.1. Botkit	12
4.2. Lenguaje de implementación	13
4.3. Motor SQL para consultas. AlaSQL	14
5. Casos de estudio	17
5.1. Ejemplo 1: Notas de asignaturas impartidas por un profesor	18
5.1.1. Hoja de cálculo con los datos	18
5.1.2. Análisis de preguntas	19
5.1.3. Definición del DSL	19
5.1.4. Ejemplo de uso del chatbot	21
5.2. Ejemplo 2: Calendario de sesiones en un congreso científico	22
5.2.1. Hoja de cálculo con los datos	22
5.2.2. Análisis de preguntas	23
5.2.3. Definición del DSL	24
5.2.4. Ejemplo de uso del chatbot	24
5.3. Ejemplo 3: Búsqueda de restaurantes de Tripadvisor	26
5.3.1. Hoja de cálculo con los datos	27
5.3.2. Análisis de preguntas	27
5.3.3. Definición del DSL	28
5.3.4. Ejemplo de uso del chatbot	29
6. Conclusiones	33
Anexos	
A. import.io: Extracción de datos tabulares a partir de la web	37
B. wit.ai: Procesamiento del lenguaje natural orientado a bots conversacionales	39

Bibliografía	41
Agradecimientos	43

Índice de figuras

2.1. Un chatbot que informa sobre las elecciones españolas del 26-J ofrecien- do información personalizada a cada usuario.	5
2.2. El chatbot define las posibles opciones que ofrece para responder median- te botones en lugar de esperar un mensaje textual.	6
3.1. Modelo de características de SheetChat.	10
4.1. Una conversación con Botkit sobre Slack.	13
5.1. Hoja de cálculo del profesor para la asignatura de matemática.	18
5.2. Hoja de cálculo del profesor para la asignatura de física.	19
5.3. DSL con la definición de las Sheet y un intent a utilizar en el ejemplo de notas.	20
5.4. Definición de la pregunta para obtener la nota ponderada de los ejercicios de física del ejemplo de notas.	21
5.5. Interacciones del usuario a la hora de consultar las notas de Alberto.	22
5.6. El chatbot recomienda algunos nombres en caso de que no se defina o se defina un nombre inexistente para hacer el filtrado en la hoja de cálculo.	23
5.7. Hoja de cálculo con el programa del WISE 2015.	23
5.8. Implementación del chatbot de la conferencia. Cabe destacar remarcado en azul el intent con input multicolumna.	25

5.9. Interacción del usuario a la hora de preguntar por los eventos en una hora concreta (Imagen izquierda y central) y consulta respecto a un topic concreto (Imagen derecha).	25
5.10. Búsqueda de restaurantes de Miami en Tripadvisor	26
5.11. Hoja de cálculo con restaurantes de Miami extraídos de Tripadvisor. . . .	27
5.12. DSL de SheetChat que describe el intent para la búsqueda por tipo de comida de los restaurantes de Tripadvisor.	28
5.13. DSL de SheetChat que describe el intent para la búsqueda por precio máximo (resaltado en azul) y tipo de comida de los restaurantes de Tripadvisor. .	29
5.14. Interacción entre el usuario y el chatbot que recomienda restaurantes en base a un tipo de cocina.	30
5.15. El bot de Tripadvisor recomienda restaurantes italianos con precio menor a 15 dolares por petición del usuario.	31

Indice de tablas

1. CAPÍTULO

Introducción

La inclusión del Smartphone se ha extendido hasta el punto de ser una herramienta indispensable en el día a día, tanto para comunicación [Montag et al., 2015], como para la búsqueda de información [Wang et al., 2016]. El uso del Smartphone en estos aspectos está superando a los sistemas de cómputo tradicionales como el PC o los portátiles. El Smartphone dispone actualmente una capacidad de trabajo similar a los PC, con la ventaja de la movilidad que ofrece. En la actualidad, con un Smartphone se pueden realizar la mayoría de tareas cotidianas que un usuario puede requerir, como leer el correo electrónico, comunicarse con sus seres queridos, consultar información en la web o realizar compras online.

Como se ha comentado previamente, el uso del Smartphone ha proliferado en los últimos años, donde su característica principal es la movilidad que ofrece frente a los PC o portátiles tradicionales. Para ofrecer esta movilidad una de las características más afectada es la del tamaño del dispositivo. Se ha pasado de las pantallas mayores de 15 pulgadas a dispositivos que llegan a un máximo de 7" (los conocidos como phablets¹).

Sin embargo, a pesar de que se puedan realizar tareas complejas, sus limitaciones provoca que algunas tareas puedan ser realmente tediosas, o incluso, imposibles de realizar. Un ejemplo claro es la consulta de información de datos en hojas de cálculo. En la actualidad el uso de hojas de cálculo como Microsoft Excel o Google Spreadsheet es una de las herramientas más utilizadas en el manejo de información, en el ámbito empresarial, pero

¹Los phablet son dispositivos móviles denominados de esta manera por comprenderse en un tamaño mayor que los smartphones (hasta 5") y menor que los tablets (a partir de 7"): <https://en.wikipedia.org/wiki/Phablet>

también a nivel personal. La potencia y versatilidad que ofrece es de sobra conocida, de ahí que exista gran cantidad de hojas de cálculo para el almacenamiento de datos. Actualmente 1 de cada 7 habitantes en el mundo utilizan alguna herramienta de hojas de cálculo, para almacenar información, pero también para consultarla.

2. CAPÍTULO

Introducción a los chatbots

En este capítulo se tratará de abordar uno de los conceptos fundamentales en los que se basa el trabajo realizado. No es otro que los chatbots. En el Apartado 2.1 se tratará de describir qué es un chatbot y en qué contextos se utiliza. En el Apartado 2.2 se explicará las actuales plataformas para el desarrollo de chatbots, sus características y la solución que se ha adoptado en este trabajo.

2.1. Definición y contexto de uso

Existen múltiples definiciones para describir qué es un chatbot, agente conversacional, o simplemente bot. Un chatbot se puede describir como un software para automatizar infinidad de tareas que actualmente desarrollan los usuarios por sí mismos, como reservar un restaurante para cenar, añadir un evento al calendario u obtener información [Wagner, 2016]. También se les describe como software con características de inteligencia artificial que pueden hacer cualquier cosa, enseñar, jugar, buscar, recordar, conectar, integrar con otros servicios,...¹.

Un concepto que habitualmente se confunde o se solapa (en cierta medida) con los chatbots son los asistentes personales o asistentes digitales. Ambas ideas comparten algunas características que se presentan a continuación:

¹¿Qué es un chatbot de telegram? <https://telegram.org/blog/bot-revolution>

- Tanto los chatbots como los asistentes personales disponen (o pueden disponer) de interfaz textual o por voz.
- Ambas ideas tienen como objetivo automatizar tareas cotidianas.
- Son capaces de integrar múltiples servicios.

Sin embargo, presentan una diferencia principal. Un chatbot está orientado a resolver ciertos problemas, digamos que es experto en un ámbito concreto, puede actuar como representante de una empresa o un servicio. Mientras tanto un asistente personal juega el papel de oráculo y tiene que lidiar con cualquier tipo de tarea [Wright, 2016].

Lo que las definiciones dejan claro es que un bot conversacional tiene un propósito específico y que en líneas generales es un software que debe de lidiar con o resolver tareas cotidianas del usuario.

Sin embargo, lo mejor para conocer el contexto, el tipo de respuestas que proporciona, las características o “habilidades” de las que dispone un agente conversacional es ver un ejemplo. En la Figura 2.1 se puede observar la interacción que tiene un usuario con un chatbot llamado Politibot. Politibot² es un agente conversacional que permite ofrecer información acerca de las elecciones generales de España. En la imagen izquierda Figura 2.1 se puede ver cómo el bot recaba cierta información del usuario (rango de edad, localización) para de esta manera ofrecer información más relevante para el elector. En la imagen derecha, el usuario pregunta por los resultados de las elecciones, a lo que el chatbot proporciona una respuesta personalizada, mostrando los datos de las elecciones generales, pero también los de su provincia.

Detrás de un chatbot de estas características se aúnan diferentes servicios: bases de datos con las preferencias de los usuarios, sistemas de localización o datos extraídos de servicios web para las noticias o los resultados.

Cualquiera puede pensar que los bots son un concepto nuevo dentro de las tecnologías de información. Sin embargo, es una idea que lleva desde los comienzos de la informática (uno de los pioneros fue el proyecto ELIZA³), aunque en los últimos años está en crecimiento [Ferrara et al., 2016]. Esto es debido al aumento de las redes sociales y de los dispositivos interconectados dentro de lo que se conoce cómo el Internet de las Cosas⁴.

²Sitio web de @politibot: <https://politibot.es/>

³Proyecto ELIZA, un chatbot que simulaba a una psicóloga: <https://en.wikipedia.org/wiki/ELIZA>

⁴Utilización de chatbots como interfaces para el Internet de las cosas: <https://iot.telefonica.com/blog/using-smart-chatbots-as-an-iot-interface>

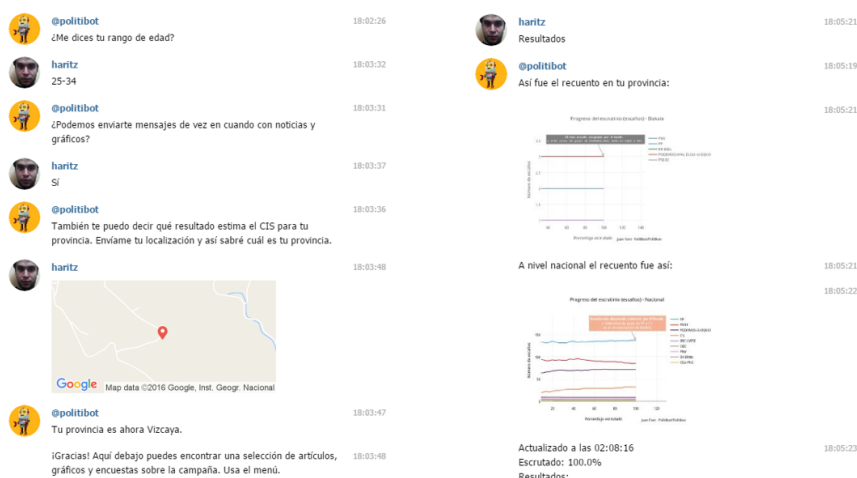


Figura 2.1: Un chatbot que informa sobre las elecciones españolas del 26-J ofreciendo información personalizada a cada usuario.

Los chatbots proporcionan una interfaz de comunicación en la que se reduce el coste frente a la interacción humana que tenían que proporcionar las empresas [Dans, 2016]. De igual manera también porque esta interacción ejerce menor presión en el usuario que quiere realizar consultas. Un bot está disponible para atender consultas 24 horas al día los 7 días de la semana, y puede atender simultáneamente consultas de múltiples usuarios, a diferencia de los humanos.

A pesar de esta reducción de recursos humanos se intensifica a la hora de crear un chatbot. Dicho de otra manera **¿Cuán sencillo es realizar un bot que satisfaga de manera adecuada las necesidades de un usuario?** Como se ha mencionado detrás hay un sistema complejo lleno de un ecosistema de tecnologías que hay que aprender y que requiere de gran conocimiento. En el próximo Apartado 2.2 se hará hincapié en las plataformas que ofrecen los chatbots y las características de los mismos.

2.2. Plataformas para desarrollo de agentes conversacionales

Como se ha mencionado en el apartado anterior, los chatbot existen desde hace varias décadas. Sin embargo, con el uso de las tecnologías móviles y el aumento del uso de aplicaciones de chat para conversar [Montag et al., 2015], ha hecho que los bots se hayan puesto en boga nuevamente.

Hay que diferenciar dos aspectos a la hora de hablar de plataformas para los chatbots. Por un lado existen las plataformas donde tiene el chatbot su interfaz, es decir, en qué aplicación o servicio mediante el cual chatea el usuario con el bot, que será en la que se centra este apartado. Por otro lado está la plataforma de desarrollo de los bots, que es la librería o servicio que se utiliza para desarrollar un chatbot que después será desplegado en una o más plataformas.

En referencia a la interfaz de los chatbots, en la actualidad muchas empresas ofrecen su plataforma como interfaz para interactuar con los chatbots. Entre ellas destacan: Facebook, Twitter, Telegram, Microsoft Skype o Slack. Sin embargo, empresas como Kik llevan trabajando años en el area de los chatbots ⁵.

La plataforma de desarrollo están muy ligada a la interfaz. Volviendo al ejemplo de politibot, en la Figura 2.2 se observa como la interfaz de Telegram proporciona botones con las diferentes opciones que el usuario puede elegir para interactuar con el bot. Esta característica es particular de Telegram, que por ejemplo Slack no dispone. Sin embargo, otras plataformas ofrecen otras características que Telegram no contempla.

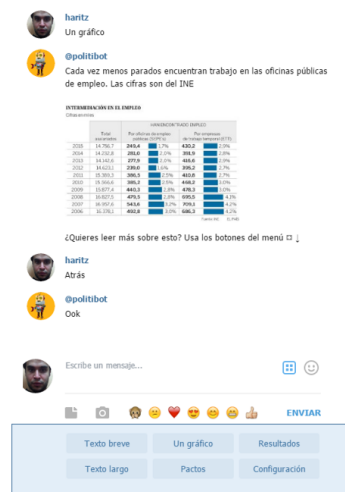


Figura 2.2: El chatbot define las posibles opciones que ofrece para responder mediante botones en lugar de esperar un mensaje textual.

Es por ello que existen muchas plataformas de desarrollo. Algunas de ellas, como Microsoft Bot Framework ⁶ ofrecen soporte multiplataforma (Telegram, Slack, Skype, Messen-

⁵¿Como predijo Kik el crecimiento del uso de los chatbots? <https://backchannel.com/how-kik-predicted-the-rise-of-chat-bots-2eaf9027b86e>

⁶Microsoft Bot Framework: <https://dev.botframework.com/>

ger,...), otras como la API de Telegram⁷ es exclusiva para Telegram.

En este trabajo, se ha decidido trabajar con la librería Botkit⁸. Las principales razones son las siguientes:

- Es multiplataforma, actualmente soporta Slack, Facebook Messenger⁹ y Twilio IP Messaging¹⁰.
- Es software libre, lo que permite ver el código fuente y modificarlo, además de que no tiene ningún coste económico.
- Es sencillo de desarrollar, permite abstraer bastante la implementación a bajo nivel de los chatbots, que es compleja debido a las llamadas asíncronas y las conexiones a múltiples servicios que trabaja por debajo.

Para profundizar en las características de implementación sobre Botkit es conveniente revisar el Capítulo 4.

⁷Telegram bot API: <https://core.telegram.org/bots>

⁸Sitio web de Botkit: <https://github.com/howdyai/botkit>

⁹Sitio web de Facebook Messenger: <https://facebook.com/>

¹⁰Sitio web de Twilio: <https://www.twilio.com/docs/api/ip-messaging>

3. CAPÍTULO

Análisis y diseño de la solución

En este capítulo se hablará del análisis y del diseño adoptado para resolver el problema de generar chatbots. Para ello, como se ha mencionado previamente, se ha definido un artefacto llamado SheetChat. En el Apartado [3.1](#) se hablará de los requisitos que ha de tener el artefacto que permita generar bots. En el Apartado [3.2](#) se realizará el análisis de las funcionalidades que ha de tener SheetChat. Finalmente, en el Apartado [3.3](#) se mostrará cual es el modelo de datos a definir para la generación de agentes conversacionales basados en hojas de cálculo.

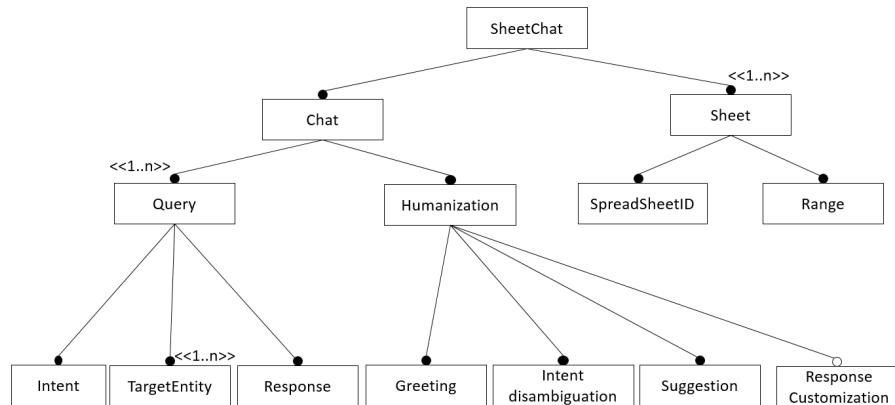


Figura 3.1: Modelo de características de SheetChat.

3.1. Requisitos

3.2. Modelo de características

3.2.1. Sheet

3.2.2. Chat

Consultas

Humanización

3.3. Modelo de datos

3.3.1. Metamodelo

3.3.2. Sintaxis concreta

4. CAPÍTULO

Implementación

En este capítulo se abordará los aspectos de implementación de SheetChat, que es un software que permite interpretar el DSL definido en el Capítulo 3 y desplegar un chatbot sobre la plataforma Slack.

Para abordar los aspectos de implementación, el capítulo se dividirá en tres secciones o apartados principales. En el Apartado 4.1 se presentará la plataforma de desarrollo de bots utilizada, Botkit, desarrollada en Javascript. En el Apartado 4.2 se hablará del lenguaje de programación utilizado y del ecosistema utilizado. Por último, en el Apartado 4.3 se hablará del motor SQL utilizado para permitir al DSL el uso de consultas enriquecidas con funciones matemáticas que se ha comentado en el Apartado 3.2.2.

De igual manera se han utilizado otros servicios web para que forman parte del funcionamiento de SheetChat, pero de los que no se hablará en este capítulo. Estos son Wit.ai¹ y la API de Google Spreadsheets². Wit.ai se utiliza en la desambiguación de intents y reconocimiento de entidades, explicado en el Apartado 3.2.2. Sobre Wit.ai se puede profundizar en el Anexo B. La API de Google Spreadsheets permite la obtención de los datos de forma matricial de la hoja de cálculo alojada en Google Docs.

Cabe destacar que el código fuente y los ejemplos del Capítulo 5 están accesibles en el siguiente repositorio de Github: <https://github.com/haritzmedina/Sheetbot>.

¹Sitio web de Wit.ai: <https://wit.ai>

²Sitio web de la documentación de la API de Google SpreadSheet: <https://developers.google.com/sheets/>

4.1. Botkit

Botkit, es una librería que permite la programación de manera más sencilla de Chatbots distribuida bajo licencia MIT. Al igual que otras plataformas mencionadas en el Apartado 2.2, como Microsoft Bot Framework o la API de bots de Telegram, Botkit permite la abstracción de algunos aspectos de programación. Botkit es una librería desarrollada por Howdy.ai. Está programada en Javascript y funciona sobre Node.js ³.

Botkit permite desarrollar bots para Slack, Facebook Messenger y Twilio. Como se ha mencionado previamente, en este trabajo se ha focalizado su uso sobre Slack. Para su uso simplemente será necesario crear un nuevo bot en Slack (o como se llama en Slack, una App ⁴) e indicarle el token de acceso al bot.

Botkit presenta un sistema relativamente sencillo para crear bots conversacionales. Se basa en una programación orientada a eventos, de ahí que case perfectamente con Javascript, que es un lenguaje dirigido por eventos. Los tipos de eventos que se pueden definir son los siguientes:

- **Escuchar mensajes:** se puede establecer que Botkit esté a la escucha de un mensaje concreto y asociar una funcionalidad en caso de que se reciba el mensaje que se desea recibir. Se puede establecer en qué canales se desea escuchar dependiendo de la plataforma. En Slack los canales son: mensaje privado, mención a él en un grupo (donde esté añadido como miembro el bot) o mensaje recibido por cualquier canal.
- **Responder/Enviar mensajes:** una funcionalidad habitual cuando se recibe un mensaje es contestar con otro. En slack se puede definir un mensaje de respuesta más o menos elaborado, con emoticonos, textos coloreados, etc.

Botkit además de gestionar recepción y envíos de mensajes es capaz de entablar conversaciones. De esta manera se puede mantener la información intercambiada durante la conversación ya que puede ser relevante para dar una respuesta más precisa. En la Figura 4.1 se puede observar un ejemplo de conversación. En este caso el usuario ha pedido una pizza y el bot le va preguntando por el tipo de pizza, el tamaño y dónde debe de entregarse. Estos tres datos van asociados al mismo pedido, por lo que puede ser interesante utilizar la noción de conversación que Botkit proporciona.

³Node.js, es un entorno de ejecución de Javascript multi-plataforma: <https://nodejs.org/en/>

⁴Creación de apps o bots en Slack: <https://api.slack.com/bot-users>

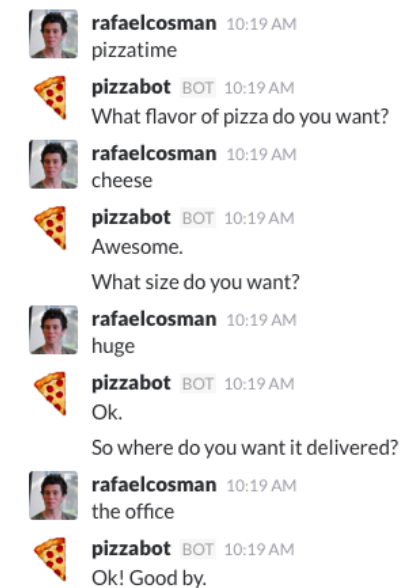


Figura 4.1: Una conversación con Botkit sobre Slack.

En la generación de bots mediante SheetChat se ha utilizado esta característica de entablar conversación también. La creación de una conversación permite ir recabando las entidades necesarias para hacer el filtrado sobre los datos de la tabla y proporcionar el resultado que el usuario necesita. De igual manera, Botkit proporciona mecanismos de repetición de preguntas para el caso de que no haya podido interpretar la información que el usuario le ha proporcionado.

Como se ha mencionado previamente, en Botkit el funcionamiento reside en que necesita escuchar un mensaje para poder comunicar algo. Para poder decidir si la conversación que hay que abrir pertenece a un Intent o a otro se ha utilizado la técnica de *Intent Disambiguation* (ver Apartado 3.2.2) mediante la herramienta de procesamiento del lenguaje natural Wit.ai. Para aunar el uso de Botkit con el de Wit.ai existe un middleware que se ha tenido que adaptar para este trabajo llamado [botkit-middleware-witai](#).

4.2. Lenguaje de implementación

El lenguaje de implementación utilizado es Javascript. En gran medida impuesto por la librería en la que se apoya SheetChat y que se acaba de exponer, Botkit, pero también porque es un lenguaje adecuado para el desarrollo de agentes conversacionales. En parti-

cular se ha utilizado el estándar ECMAScript2015⁵ o ECMA 6, que viene a ser la nueva versión de Javascript.

Tal y como se ha mencionado antes, el desarrollo se ha realizado sobre Node.js. Cabe destacar que los bots se ejecutan de manera local y se conectan a las diferentes APIs de los servicios de mensajería instantánea, como puede ser en este caso Slack. La idea es que esta aplicación resida en un servidor dedicado que permita su ejecución dando disponibilidad al bot de manera continua.

4.3. Motor SQL para consultas. AlaSQL

Una de las piezas fundamentales y sobre las que se basa la teoría de este proyecto son las hojas de cálculo y las consultas y filtrados que se le pueden realizar. Google SpreadSheet dentro de su API no proporciona filtrados de ningún tipo, ni un lenguaje potente para realizar consultas específicas sobre los datos. Es por ello que se ha decidido utilizar una implementación de SQL para javascript, llamada AlaSQL ⁶.

El método de trabajo con AlaSQL en SheetChat es sencillo. Las filas y columnas que se reciben de Google SpreadSheet se traducen a tablas relacionales, cada tabla está representada por los valores de una hoja de cálculo. En caso de que los datos se actualicen, estos serán actualizados en la tabla de la base de datos en memoria de AlaSQL.

AlaSQL permite el almacenamiento en ficheros, importar datos de CSV, etc. Aunque en este caso sólo se ha utilizado el almacenamiento en memoria de los datos de las hojas de cálculo. Esto reduce en gran medida el tamaño de los datos que un Chatbot puede tener, pero resulta suficiente para un Chatbot de auto-consumo, y la velocidad de consulta es mucho mayor.

Cuando un usuario realiza una petición de unos datos al chatbot, estas consultas se traducen a sentencias SQL que serán ejecutadas sobre AlaSQL. Basándonos en el modelo de características de la Figura 3.1, se utiliza el response para definir las columnas a mostrar, la hoja de cálculo del intent como la tabla de la base de datos y las entidades como filtros. Suponiendo que la respuesta que da el Chatbot es “Examen1, Examen2, Examen3”, la hoja de cálculo es “Notas” y la entidad requerida es “Alumno”, la consulta se traduciría a algo similar a esta sentencia SQL: *SELECT Examen1, Examen2, Examen3 FROM Notas WHERE Alumno= 'X'*.

⁵Sitio web de ECMAScript2015: <http://www.ecma-international.org/ecma-262/6.0/>

⁶AlaSQL, una implementación de base de datos SQL en memoria en javascript: <http://alasql.org/>

De esta manera se pueden realizar consultas más elaboradas, con funciones típicas de SQL como SUM, AVG, MAX, LIKE,... siempre que se mantenga la estructura de la consulta. Se descartó la idea de permitir al usuario que pudiese definir sus propias sentencias SQL limitándolas a la estructura SELECT-FROM-WHERE por las siguientes razones:

- La idea, de cara al futuro, es poder generar la sintaxis concreta de SheetChat mediante una herramienta gráfica que haga mucho más sencilla la especificación de los bots y no tenga el usuario que aprender el DSL.
- La idea no es consultar datos que requieran un procesamiento muy complejo, ya que para el procesamiento de fórmulas matemáticas complejas ya está la propia hoja de cálculo. El objetivo es poder prestar unas entidades derivadas de manera sencilla.

5. CAPÍTULO

Casos de estudio

El diagrama de características estudiado previamente se ha de resolver en la definición de un chatbot. En este apartado se estudiará demostrará el funcionamiento del DSL definido para la creación de un SheetChat. Para ello se tienen que tener en cuenta todas las características previamente mencionadas, la definición del origen de datos, la creación de una conversación que nos proporcione una petición de datos de entrada y su salida asociada; y no menos importante, los recursos que nos permitan humanizar el bot para ofrecer una experiencia de usuario agradable.

Para ello en este trabajo se han elaborado tres ejemplos. El primero de ellos es dado las notas de dos asignaturas impartidas por un profesor de primaria, el poder preguntar por las notas de los alumnos que tiene. El segundo de los ejemplos permite dado un calendario de sesiones de un congreso científico, en este caso extraído del WISE de 2015¹, poder obtener información sobre qué sesiones hay en los diferentes slots (u horarios). Por último, el tercer ejemplo permite, dada una hoja de cálculo autogenerada de una búsqueda de restaurantes de Miami en el sitio web Tripadvisor, obtener restaurantes por tipo de comida.

Los tres casos de estudio tendrán la misma estructura. En primer lugar se abordará el problema que el usuario tiene. En segundo lugar se realizará un análisis de las preguntas o cuestiones que tendrá que ser capaz de resolver el chatbot. Posteriormente se hará hincapié en la definición del DSL que tendrá que hacer el usuario para resolver su problemática.

¹Calendario con las diferentes sesiones del congreso WISE: <http://www4.cis.fiu.edu/wise2015/@schema.html>

Finalmente se mostrará un ejemplo de interacción entre el chatbot generado y el usuario.

5.1. Ejemplo 1: Notas de asignaturas impartidas por un profesor

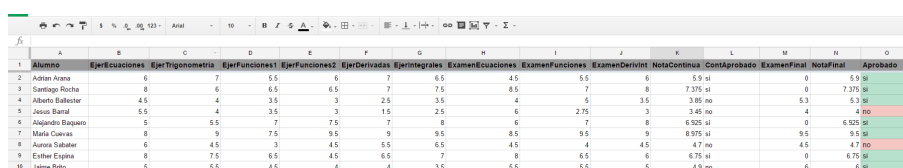
Para el primer ejemplo la audiencia objetivo es un profesor de instituto que dispone de las notas de sus alumnos almacenadas en hojas de cálculo. Él es profesor de dos asignaturas Matemática y Física.

5.1.1. Hoja de cálculo con los datos

Para el almacenamiento de las notas de sus alumnos dispone de dos hojas de cálculo, una con las notas de Matemática (ver Figura 5.1) y otra para las calificaciones de la asignatura de Física (ver Figura 5.2).

Tal y como se puede observar en la Figura 5.1 el profesor tiene las notas de cada uno de sus alumnos que va rellenando a medida que se les evalúa de los diferentes aspectos de las asignaturas. Por lo tanto, cada fila representa a un alumno y cada columna a cada concepto a evaluar de la asignatura.

En el método de evaluación empleado por el profesorado se tiene en cuenta trabajos de clase o ejercicios y exámenes durante la evaluación continua. En caso de que la media ponderada de estos parciales supere un 5 el alumno habrá aprobado mediante la evaluación continua. En caso contrario, tendrá que presentarse a un examen final con todo el temario del curso. A modo de ejemplo se puede visualizar en la Figura 5.1 como el alumno Adrian Arana ha aprobado la evaluación continua con un 5.9 mientras que Alberto Ballester tuvo que presentarse al examen final para aprobar dado que en la evaluación continua su nota era de 3.85.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Alumno	EjerEcuaciones	EjerTrigonometria	EjerFunciones1	EjerFunciones2	EjerDerivadas	EjerIntegrales	ExamenEcuaciones	ExamenFunciones	ExamenDerivadas	NotaContinua	ContAprobado	ExamenFinal	NotaFinal	Aprobado
2	Adrian Arana	6	7	5.5	6	7	6.5	4.5	5.5	6	5.9 si		0	5.9 si	
3	Santiago Rocha	8	6	6.5	6.5	7	7.5	8.5	7	8	7.375 si		0	7.375 si	
4	Alberto Ballester	4.5	4	3.5	3	2.5	3.5	4	5	3.5	3.85 no		5.3	5.3 si	
5	Jesus Baral	5.5	4	3.5	3	1.5	2.5	6	2.75	3	4 no		4	4 no	
6	Alejandro Baquero	5	5.5	7	7.5	7	8	6	7	8	6.925 si		0	6.925 si	
7	Maria Cuervas	8	9	7.5	9.5	9	9.5	8.5	9.5	9	8.975 si		9.5	9.5 si	
8	Aurora Salazar	6	4.5	3	4.5	5.5	6.5	4.5	4	4.5	4.7 no		4.5	4.7 no	
9	Esther Espina	8	7.5	6.5	4.5	6.5	7	8	6.5	6	6.75 si		0	6.75 si	
10	Jaime Brito	5	5.5	4.5	4	4	3.5	5.5	5.5	5	4.9 no		6	9 si	

Figura 5.1: Hoja de cálculo del profesor para la asignatura de matemática.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Alumno	EjerCinematica	EjerDinamica	EjerEnergia	EjerGravitatorio	Examen1	Examen2	NotaContinua	ContinuaAprobado	ExamenFinal	NotaFinal	Aprobado
2	Adrian Arana	5	5	5.5	6.5	7	6.5	6.25	si	0	6.25	si
3	Santiago Rocha	8	7.5	7	6.5	6	7	6.775	si	0	6.775	si
4	Alberto Ballester	5	4	3.5	4	5	5	4.6	no	5.5	5.5	si
5	Jesus Barral	3.5	3	4.5	0	4	4.5	3.625	no	4	4	no
6	Alejandro Baquero	5	6.5	6	7	6.5	6	6.275	si	0	6.275	si
7	Maria Cuevas	8	7.5	7	8.5	8.5	9	8.325	si	0	8.325	si
8	Aurora Sabater	1.5	3.5	3	4.5	4.5	4	3.9	no	5.5	5.5	si
9	Esther Espina	7.5	6	5.5	4	5.5	4.5	5.225	si	0	5.225	si
10	Jaime Brito	7	7.5	7	8.5	7.5	9	7.975	si	8.5	8.5	si
11												
12												

Figura 5.2: Hoja de cálculo del profesor para la asignatura de física.

5.1.2. Análisis de preguntas

Una vez conocidos los datos almacenados, el profesor que quiere generar un chatbot tiene que tener en cuenta qué tipo de análisis quiere hacer sobre los datos, o dicho de otra manera, qué consultas va a realizar sobre los datos. Debería de ser capaz de decidir cuáles son las preguntas más habituales sobre esos datos o las que más frecuentemente le piden sus alumnos. Dado que son asignaturas diferentes y su método de evaluación no es exactamente el mismo, las preguntas que se puede llegar a realizar también son diferentes.

A continuación se recogen algunas de las preguntas que podría hacer el profesor:

- **¿Cuál es la nota que tiene un alumno en cada uno de los ejercicios de mate/física?** Es habitual preguntar por la nota en uno de los ejercicios o en el conjunto de los mismos a lo largo de la evaluación.
- **¿Cuál es la nota de un alumno en los exámenes parciales de mate/física?** Al igual que sucede con los ejercicios, es interesante conocer las notas de los exámenes parciales.
- **¿Se debe de presentar un alumno al examen final de mate?** O dicho de otra manera, ¿ha aprobado la evaluación continua?
- **¿Qué nota debe de sacar un alumno en los exámenes de física para aprobar la asignatura?** O preguntado de otra manera, ¿Qué nota ponderada tiene en los ejercicios de evaluación continua?
- **¿Ha aprobado un alumno la asignatura de física?** De esta manera se sabe si un alumno ha de presentarse a la segunda convocatoria de física o no.

5.1.3. Definición del DSL

Como se ha comentado en el Capítulo 3 hay que definir mediante el uso del DSL de SheetChat el origen de los datos y los intents para cada una de las consultas que se quiera

extraer de la hoja de cálculo previamente definidas en el Apartado 5.1.2.

Si se observa la Figura 5.3 se puede observar cómo es la definición de las hojas de cálculo a consultar. De igual manera se define una descripción para el chatbot y un mensaje de bienvenida que nos ayude a recordar alguna de las funcionalidades del bot. Posteriormente se muestra un intent de los que deberá de definir el profesor que vaya a utilizar el chatbot.

El intent a resolver es el de obtener la nota de los ejercicios de mate de un determinado alumno. El origen de datos por tanto será la hoja de NotasMate. Se define que las columnas que el bot debe de responder son las de los ejercicios. También se define que sólo se desea un resultado, ya que sólo se pregunta por un alumno concreto. Esta característica es útil para mostrar un número determinado de resultados. En este caso cada alumno se identifica inequívocamente por el nombre y apellido. También se indica que se muestre el nombre de la columna (en este caso el nombre del ejercicio).

Dentro del intent también se definen las entidades de filtrado. En este caso se define la columna Alumno. El tipo de filtrado utilizado es LIKE, que tiene la misma funcionalidad que el like de SQL. Esto permite no tener que introducir nombre y apellido del alumno, si no introducir parcialmente su nombre a la hora de filtrar. En lugar de exigir al usuario escribir Adrian Arana, podrá preguntar por las notas de Adrian o de Arana. Por último se definen mensajes que permitan al usuario preguntar de una manera más amigable (o humana, ver Apartado 3.2.2) por las entidades que hacen falta.

<pre> "sheets": [{ "sheetName": "NotasMate", "sheetRange": "A1:O10" }, { "sheetName": "NotasFisica", "sheetRange": "A1:L10" }], "chat": { "description": "Este bot ayuda a obtener las notas de un alumno.", "greetings": "Hola! Soy notabot y te ayudaré a obtener tus calificaciones. Prueba preguntandome por tus notas de matematica o fisica", "intents": [{ "id": "notasMateEjerciciosPorAlumno", "sourceSheet": "NotasMate", "responses": { "outputColumn": "EjerEcuaciones, EjerTrigonometria, EjerFunciones1, EjerFunciones2, EjerDerivadas, EjerIntegrales", "numberOfResponses": 1, "notFoundMessage": "No se han encontrado trabajos para el alumno.", "showColumnName": true }, "entities": [{ "inputColumn": "Alumno", "mask": "LIKE", "entityMissingMessage": "Dime el nombre del alumno para obtener las notas de ejercicios de matemática. Como por ejemplo:", "entityNotFoundMessage": "No tengo información sobre ese alumno. Podrias preguntar por los ejercicios de mate de" }] }] } </pre>	<p>Hoja de cálculo</p> <p>Saludo</p> <p>Intent</p> <p>Respuesta</p> <p>Entidades</p>
--	--

Figura 5.3: DSL con la definición de las Sheet y un intent a utilizar en el ejemplo de notas.

Algunas preguntas son más difíciles de formular debido a la naturaleza de los datos. Sin embargo, SheetChat ofrece algunos mecanismos que permiten resolver funciones matemáticas a la hora de realizar consultas sobre los datos. Se puede observar en la pregunta relacionada con obtener la nota ponderada. La nota ponderada de los ejercicios de Física

el profesor lo tiene definido de una manera determinada. Esto no es una columna como tal dentro de la hoja de cálculo, si no que es una entidad derivada. En la Figura 5.4 se observa que dentro del response hay una expresión matemática que define qué es la nota ponderada de los ejercicios. De igual manera, en las respuestas se puede proporcionar un mensaje personalizado para mejorar la humanización del bot ofreciendo una respuesta más sencilla de interpretar por un humano.

```
{
  "ID": "fisicaPonderadaEjerciciosPorAlumno",
  "sourceSheet": "NotasFisica",
  "response": {
    "outputColumn": "EjerCinematica*0.05+EjerDinamica*0.15+EjerEnergia*0.1+EjerGravitatorio*0.1 AS Ponderada, Alumno",
    "numberOfResponses": 1,
    "notFoundMessage": "No se han encontrado resultados para el alumno.",
    "customResponseStructure": ["El alumno %s tiene una nota ponderada en ejercicios de fisica de %s sobre 4 puntos.", "Alumno", "Ponderada"]
  },
  "entities": [
    {
      "inputColumn": "Alumno",
      "mask": "LIFE",
      "entityMissingMessage": "¿De qué alumno quieres obtener la nota ponderada de ejercicios de fisica? Estos son algunos alumnos que tengo:",
      "entityNotFoundMessage": "Ups! no conozco a ese alumno. Podrias preguntar por la nota de ejercicios de fisica de"
    }
  ]
}
```

Figura 5.4: Definición de la pregunta para obtener la nota ponderada de los ejercicios de física del ejemplo de notas.

5.1.4. Ejemplo de uso del chatbot

A continuación se puede visualizar cómo sería un ejemplo de interacción entre un ser humano y el bot que se ha generado a partir del DSL de SheetChat en el ejemplo de las notas. En la Figura 5.5 se puede observar un intercambio de mensajes para obtener información respecto al alumno Alberto. En primer lugar se ha saludado al chatbot para que este proporcione su mensaje de bienvenida. Esta interacción no es necesaria si se conocen cuales son las características del chatbot. Posteriormente se le han preguntado por los resultados de alberto en matemática: si habia aprobado la evaluación continua y las notas que ha obtenido para conocer la causa de su suspenso. De igual manera se han realizado dos preguntas respecto a las notas de física.

En este caso el chatbot ha inferido cuales eran los intents del usuario y a su vez la entidad necesaria para cada una de ellas, en este caso el nombre del alumno. Si se obtienen de la frase de manera adecuada los intents y las entidades el chatbot proporciona la respuesta. En caso de que se haya inferido el intent pero no las entidades, preguntará por ellas (ofreciendo sugerencias que ayuden al usuario). Esto sucede en la Figura 5.6 dado que no se le había proporcionado ningún nombre a la hora de preguntar por la media en los ejercicios de física. De igual manera, si se le proporciona un nombre no existente, el chatbot le seguirá ofreciendo sugerencias.

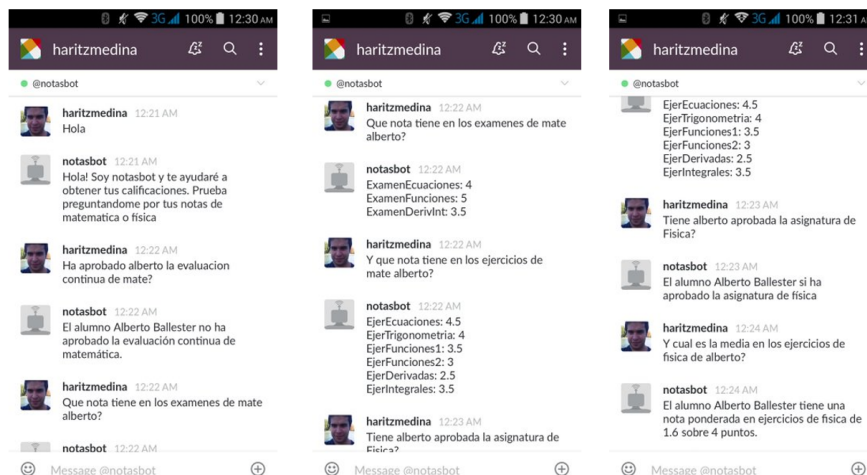


Figura 5.5: Interacciones del usuario a la hora de consultar las notas de Alberto.

5.2. Ejemplo 2: Calendario de sesiones en un congreso científico

Los congresos científicos aúnan conocimiento de distinta índole. Habitualmente estos eventos disponen de un calendario complejo, dónde hay trabajos más o menos interesantes o relevantes con la rama de especialización que tiene el asistente. Es por ello que acudir a las sesiones más afines a tu trabajo es importante. Es interesante disponer de información in situ de cuales son las próximas charlas que habrá, dónde o quién las presenta. Sin embargo, los sitios webs rara vez están preparados para su navegación por el móvil o se pierde mucho tiempo en encontrar los eventos a los que se desea acudir. Es una información relevante y que se desea conocer en el momento.

Este ejemplo se centra en ofrecer una alternativa en forma de bot mediante una hoja de cálculo creada a partir del horario de la conferencia WISE 2015 ².

5.2.1. Hoja de cálculo con los datos

Como se ha mencionado previamente la hoja de cálculo es creada a partir de una tabla con el programa disponible en el sitio web. En la Figura 5.7 se puede observar el calendario de las diferentes sesiones que están presentes en el congreso. Cada sesión tiene asociada

²Programa del WISE 2015 (Web Information System Engineering) <http://www4.cis.fiu.edu/wise2015/@schema.html>



Figura 5.6: El chatbot recomienda algunos nombres en caso de que no se defina o se defina un nombre inexistente para hacer el filtrado en la hoja de cálculo.

un día y un slot (una franja horaria) donde se exponen trabajos. En algunos de los slots se produce un solapamiento de múltiples sesiones, por lo que es interesante que el usuario conozca cuales son las charlas para elegir la más interesante.

	A	B	C	D	E	F
1	Day	Slot	Session1	Session2	Session3	Session4
2	Sunday, November 1st	7:30-9:00	Room: Paris Breakfast			
3	Sunday, November 1st	9:00-10:00	Room: Paris Chair: S.S. Venugur Keynote 1: Hardware Accelerators for the Web			
4	Sunday, November 1st	10:30-12:00	Room: Nice Chair: Jin Shao \$1:Big Data Techniques	Room: Cannes Chair: Devis Blanchini \$2:Social Issues of Web Applications	Room: Marseille Chair: Yanchun Zhang \$3:Devo/Hidden Web	Room: Paris Chair: Chen Chen & William Wei Song \$51:Data Quality and Trust in Big Data
5	Sunday, November 1st	12:00-13:30	Room: Paris Lunch			
6	Sunday, November 1st	13:30-15:00	Room: Nice Chair: Dingding Wang \$4:Big Data and Mining Techniques	Room: Cannes Chair: Marek Rusinkiewicz \$5:Social Network Computing	Room: Marseille Chair: Eduardo B. Fernandez \$6:Secure Web Systems Architectures Using Secu	Room: Paris Chair: Shaohong Zhang \$52:Data Quality and Trust in Big Data
7	Sunday, November 1st	15:30-17:00	Room: Nice Chair: Hideyuki Takada \$6:Big Data Applications	Room: Cannes Chair: Frederic Lefebvre \$7:Social Web	Room: Marseille Chair: Eduardo B. Fernandez \$6:Secure Web Systems Architectures Using Secu	Room: Paris Chair: Sita Yella \$53:Data Quality and Trust in Big Data
8	Sunday, November 1st	17:30-18:30	Room: Paris Reception			
9	Monday, November 2nd	7:30-9:00	Room: Paris Breakfast			
10	Monday, November 2nd	9:00-10:00	Room: Monte Carlo Chair: Tao Li ata Science Research at FIU (S. S. Iyengar, Naphthal Rishie, Steve Luis, and Radu Jianu)			
11	Monday, November 2nd	10:30-12:00	Room: Nice Chair: Chao Gong \$8:Web Data Integration and Mashups	Room: Cannes Chair: Shu-Ching Chen \$9:Web Privacy and Security	Room: Monte Carlo Chair: Richi Navek \$10:Integration of Web with IoT	
12	Monday, November 2nd	12:00-13:30	Room: Paris Lunch			
13	Monday, November 2nd	13:30-15:00	Room: Nice Chair: Junbo Wang \$11:Web Data Models	Room: Cannes Chair: Dimitrios Theodoratos \$12:Web Information Retrieval	Room: Monte Carlo Chair: Byoung Ho Kang \$13:Web-based recommendations	
14	Monday, November 2nd	15:30-17:00	Room: Nice Chair: Toshiyuki Arimura \$14:Linked Open Data	Room: Cannes Chair: Wojciech Cellary \$15:Web Information Extraction	Room: Monte Carlo Chair: Mark A. Finlayson The WordNet Database: Form, Function, and Use	
15	Monday, November 2nd	17:00-17:30			Room: Monte Carlo Chair: Mark A. Finlayson T2:The WordNet Database: Form, Function, and Use	
16			Room: Monte Carlo Ballroom I&II			

Figura 5.7: Hoja de cálculo con el programa del WISE 2015.

5.2.2. Análisis de preguntas

Como se ha comentado previamente, el usuario del bot tiene como objetivo poder conocer el programa del congreso durante la estancia en él. Algunas de las preguntas que le pueden surgir durante el evento están recogidas a continuación:

- **¿Cuáles son las charlas en un horario concreto, es decir, qué eventos hay en esa franja horaria?** De los eventos que existan en esa franja horaria, el asistente al congreso podría decantarse por la que más interesante le resulte.
- **¿Qué eventos hay relacionados con un tema (o topic) en particular?** De esta manera el usuario sabrá con una simple pregunta en qué horarios están programadas charlas interesantes para él.
- **¿Quiero conocer en qué sesiones participa una persona como chairman?** Los investigadores conocen el trabajo de algunas personas y puede interesarle saber si estas personas participan como expertas en la materia de una determinada sesión.

5.2.3. Definición del DSL

El diseño del chatbot en este ejemplo constará de una única hoja de cálculo y de dos intents. A pesar de que previamente se han definido 3 cuestiones, sólo hace falta definir dos intents debido a que la obtención de las sesiones relacionadas con un tema en concreto o presidido por un chairman concreto se pueden unir en una. Esto es debido a la naturaleza de los datos. Una columna sesión tiene como valor un string que incluye ambas informaciones. A la hora de realizar el filtrado, se filtra por valor de la columna que se desee. Visto en un ejemplo, observando la sesión 1 del domingo en el slot 10:30-12:00, es lo mismo filtrar por el topic *Hardware Accelerators for the Web* por la habitación *Paris* o el chairman *Iyengar*, se obtendrá como resultado el día y slot de esa sesión.

Por otro lado, en este ejemplo, surge la novedad de filtrar por más de una columna. En la Figura 5.8 se observa remarcado en color azul el entity definido. En este caso se han definido cuatro columnas que actuarán como filtro. A la hora de preguntar por el topic o chairman, el chatbot buscará si se dan positivos en cualquiera de las cuatro columnas. En caso de que el valor introducido por el usuario dé positivo por alguna de las cuatro columnas, se le mostrará el día y la hora (o slot) de ese evento.

5.2.4. Ejemplo de uso del chatbot

A continuación se presenta un ejemplo de interacción real con el chatbot desarrollado para el programa de las conferencias. Para la obtención de las sesiones en una franja horaria concreta, tal y como se ha mencionado en la implementación del chatbot, se requiere de dos parámetros (el día y el slot). En la Figura 5.9 el Chatbot guía al usuario mediante

```

"sheets": [
  {
    "gsheetName": "Schedules",
    "gsheetRange": "A1:F22"
  }
],
"chat": {
  "description": "WISE conference sessions related bot.",
  "greetings": "This bot helps you to retrieve information related with the WISE 2015 conference. Try typing 'Show sessions'.",
  "intents": [
    {
      "ID": "retrieveSlotByTopic",
      "sourceSheet": "Schedules",
      "response": {
        "outputColumn": "Day, Slot",
        "notFoundMessage": "Events not found",
        "numberOfResponses": 5,
        "customResponseStructure": ["An event at %s on %s is scheduled.", "Slot", "Day"]
      },
      "entities": [
        {
          "inputColumn": ["Session1", "Session2", "Session3", "Session4"],
          "mask": "LIKE",
          "entityMissingMessage": "Search for a topic, room or chairman, for example:",
          "entityNotFoundMessage": "Topic, room or chairman not found, try another one:"
        }
      ]
    }
  ]
}

```

Figura 5.8: Implementación del chatbot de la conferencia. Cabe destacar remarcado en azul el intent con input multicolumna.

preguntas para obtener la información suficiente de a qué slot se refiere el usuario. En este caso realiza una pregunta respecto a los días y otra respecto al slot. Si el usuario hubiese añadido esa información a la hora de hacer la pregunta, el chatbot podría haber inferido estas entidades y ahorrarse los pasos de preguntarlo. Esto es lo que sucede en la pregunta de la imagen derecha, donde el usuario ha proporcionado un criterio de búsqueda.

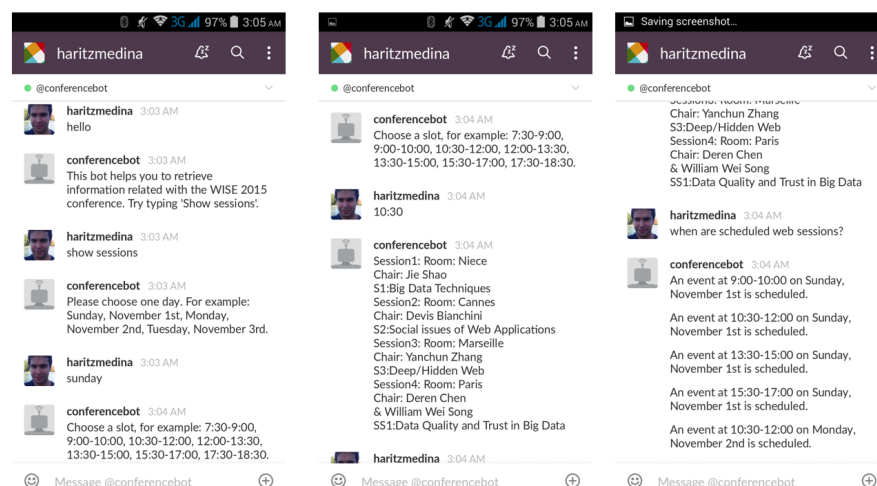


Figura 5.9: Interacción del usuario a la hora de preguntar por los eventos en una hora concreta (Imagen izquierda y central) y consulta respecto a un topic concreto (Imagen derecha).

5.3. Ejemplo 3: Búsqueda de restaurantes de Tripadvisor

En este tercer caso de estudio se muestra además de un nuevo contexto de uso, el uso de un origen de datos web. En la actualidad la mayoría de información se puede recabar en la red, sin embargo esta no suele tener una estructura utilizada en un ámbito general como son las tablas de bases de datos o las hojas de cálculo.

En este ejemplo se mostrará cómo se pueden realizar búsquedas en restaurantes extraídos del sitio web Tripadvisor. Tripadvisor es un sitio web de opinión sobre hoteles, restaurantes, y otros lugares de ocio. En su web se pueden filtrar los resultados en base a localización (como una ciudad o provincia), características del sitio (número de estrellas de un hotel o tipo de comida de un restaurante) y otros muchos aspectos. Concretamente en este ejemplo se ha realizado una búsqueda de restaurantes en Miami (Estados Unidos), que es lo que se puede observar en la Figura 5.10.

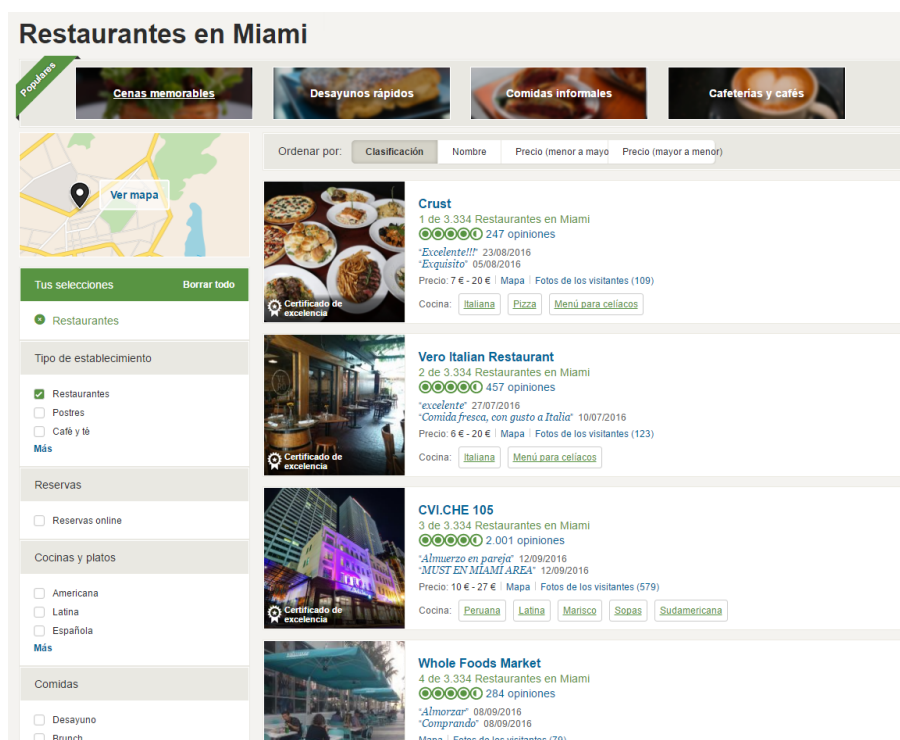


Figura 5.10: Búsqueda de restaurantes de Miami en Tripadvisor

La idea para este ejemplo es demostrar que estos datos en la web pueden ser extraídos a una hoja de cálculo y generar un chatbot mediante la herramienta SheetChat.

5.3.1. Hoja de cálculo con los datos

Como se ha mencionado al comiendo de este caso de estudio, el objetivo es poder representar estos datos en forma de hoja de cálculo. Para ello en la web existen múltiples herramientas de web scraping (extracción de información de la web). A pesar de que la web de tripadvisor muestre los restaurantes en un formato más amigable para el ser humano, si que todos los restaurantes comparten una estructura similar. En la Figura 5.10 se observa que todos los restaurantes tienen un hyperlink con el nombre del restaurante donde pinchando saldría la ficha del restaurante en cuestión. Cada restaurante tiene una imagen asociada, un número de opiniones, dos breves opiniones, el precio promedio del menú del restaurante y el tipo de comida. Precisamente, el encontrar patrones y extraer datos es el objetivo de herramientas como Import.io³.

El funcionamiento de `import.io` permite extraer una tabla con los resultados de la búsqueda de restaurantes de Miami. En el Anexo A se puede profundizar en el funcionamiento de la herramienta. Para el ejemplo, simplemente la hoja de cálculo que obtenemos es la presentada en la Figura 5.11. Mediante `import.io` se han extraído 400 restaurantes en Miami. En la tabla se pueden observar las columnas sobre las que podremos realizar preguntas posteriormente al bot, nombre del restaurante (`Name`), sitio web con la ficha de Tripadvisor (`url`), rango de precios de los menús (`RangePrice`) o tipo de comida (`Cuisines`).

A		B		C		D		E		F		G		H		I		J		K		L		M		N		O		P		Q		R		S		T		U		V		W		X		Y		Z		AA		AB		AC		AD		AE		AF		AG		AH		AI		AJ		AK		AL		AM		AN		AO		AP		AQ		AR		AS		AT		AU		AV		AW		AX		AY		AZ		BA		BB		BC		BD		BE		BF		BG		BH		BI		BJ		BK		BL		BM		BN		BO		BP		BQ		BR		BS		BT		BU		BV		BW		BX		BY		BZ		CA		CB		CC		CD		CE		CF		CG		CH		CI		CJ		CK		CL		CM		CN		CO		CP		CQ		CR		CS		CT		CU		CV		CW		CX		CY		CZ		DA		DB		DC		DD		DE		DF		DG		DH		DI		DJ		DK		DL		DM		DN		DO		DP		DQ		DR		DS		DT		DU		DV		DW		DX		DY		DZ		EA		EB		EC		ED		EE		EF		EG		EH		EI		EJ		EK		EL		EM		EN		EO		EP		EQ		ER		ES		ET		EU		EV		EW		EX		EY		EZ		FA		FB		FC		FD		FE		FF		FG		FH		FI		FJ		FK		FL		FM		FN		FO		FP		FQ		FR		FS		FT		FU		FV		FW		FX		FY		FZ		GA		GB		GC		GD		GE		GF		GG		GH		GI		GJ		GK		GL		GM		GN		GO		GP		GQ		GR		GS		GT		GU		GV		GW		GX		GY		GZ		HA		HB		HC		HD		HE		HF		HG		HH		HI		HJ		HK		HL		HM		HN		HO		HP		HQ		HR		HS		HT		HU		HV		HW		HX		HY		HZ		IA		IB		IC		ID		IE		IF		IG		IH		II		IJ		IK		IL		IM		IN		IO		IP		IQ		IR		IS		IT		IU		IV		IW		IX		IY		IZ		JA		JB		JC		JD		JE		JF		JG		JH		JI		JJ		JK		JL		JM		JN		JO		JP		JQ		JR		JS		JT		JU		JV		JW		JX		JY		JZ		KA		KB		KC		KD		KE		KF		KG		KH		KI		KJ		KL		KM		KN		KO		KP		KQ		KR		KS		KT		KU		KV		KW		KX		KY		KZ		LA		LB		LC		LD		LE		LF		LG		LH		LI		LJ		LK		LL		LM		LN		LO		LP		LQ		LR		LS		LT		LU		LV		LW		LX		LY		LZ		MA		MB		MC		MD		ME		MF		MG		MH		MI		MJ		MK		ML		MM		MN		MO		MP		MQ		MR		MS		MT		MU		MV		MW		MX		MY		MZ		NA		NB		NC		ND		NE		NF		NG		NH		NI		NJ		NK		NL		NM		NN		NO		NP		NQ		NR		NS		NT		NU		NV		NW		NX		NY		NZ		OA		OB		OC		OD		OE		OF		OG		OH		OI		OJ		OK		OL		OM		ON		OO		OP		OQ		OR		OS		OT		OU		OV		OW		OX		OY		OZ		PA		PB		PC		PD		PE		PF		PG		PH		PI		PJ		PK		PL		PM		PN		PO		PP		PQ		PR		PS		PT		PU		PV		PW		PX		PY		PZ		QA		QB		QC		QD		QE		QF		QG		QH		QI		QJ		QK		QL		QM		QN		QO		QP		QQ		QR		QS		QT		QU		QV		QW		QX		QY		QZ		RA		RB		RC		RD		RE		RF		RG		RH		RI		RJ		RK		RL		RM		RN		RO		RP		RQ		RR		RS		RT		RU		RV		RW		RX		RY		RZ		SA		SB		SC		SD		SE		SF		SG		SH		SI		SJ		SK		SL		SM		SN		SO		SP		SQ		SR		SS		ST		SU		SV		SW		SX		SY		SZ		TA		TB		TC		TD		TE		TF		TG		TH		TI		TJ		TK		TL		TM		TN		TO		TP		TQ		TR		TS		TT		TU		TV		TW		TX		TY		TZ		UA		UB		UC		UD		UE		UF		UG		UH		UI		UJ		UK		UL		UM		UN		UO		UP		UQ		UR		US		UT		UU		UV		UW		UX		UY		UZ		VA		VB		VC		VD		VE		VF		VG		VH		VI		VJ		VK		VL		VM		VN		VO		VP		VQ		VR		VS		VT		VU		VV		VW		VX		VY		VZ		WA		WB		WC		WD		WE		WF		WG		WH		WI		WJ		WK		WL		WM		WN		WO		WP		WQ		WR		WS		WT		WU		WV		WW		WX		WY		WZ		XA		XB		XC		XD		XE		XF		XG		XH		XI		XJ		XK		XL		XM		XN		XO		XP		XQ		XR		XS		XT		XU		XV		XW		XX		XY		XZ		YA		YB		YC		YD		YE		YF		YG		YH		YI		YJ		YK		YL		YM		YN		YO		YP		YQ		YR		YS		YT		YU		YV		YW		YX		YY		YZ		ZA		ZB		ZC		ZD		ZE		ZF		ZG		ZH		ZI		ZJ		ZK		ZL		ZM		ZN		ZO		ZP		ZQ		ZR		ZS		ZT		ZU		ZV		ZW		ZX		ZY		ZZ	
1	url	Image	Image_at	Image_BA	Name	Name_IG	Ranking	NumberComments	NumberOfReviews	Review1	Review1_R	Review2	Review2_R	Review3	Review3_R	Review4	Review4_R	Review5	Review5_R	Review6	Review6_R	Review7	Review7_R	Review8	Review8_R	Review9	Review9_R	Review10	Review10_R	Review11	Review11_R	Review12	Review12_R	Review13	Review13_R	Review14	Review14_R	Review15	Review15_R	Review16	Review16_R	Review17	Review17_R	Review18	Review18_R	Review19	Review19_R	Review20	Review20_R	Review21	Review21_R	Review22	Review22_R	Review23	Review23_R	Review24	Review24_R	Review25	Review25_R	Review26	Review26_R	Review27	Review27_R	Review28	Review28_R	Review29	Review29_R	Review30	Review30_R	Review31	Review31_R	Review32	Review32_R	Review33	Review33_R	Review34	Review34_R	Review35	Review35_R	Review36	Review36_R	Review37	Review37_R	Review38	Review38_R	Review39	Review39_R	Review40	Review40_R	Review41	Review41_R	Review42	Review42_R	Review43	Review43_R	Review44	Review44_R	Review45	Review45_R	Review46	Review46_R	Review47	Review47_R	Review48	Review48_R	Review49	Review49_R	Review50	Review50_R	Review51	Review51_R	Review52	Review52_R	Review53	Review53_R	Review54	Review54_R	Review55	Review55_R	Review56	Review56_R	Review57	Review57_R	Review58	Review58_R	Review59	Review59_R	Review60	Review60_R	Review61	Review61_R	Review62	Review62_R	Review63	Review63_R	Review64	Review64_R	Review65	Review65_R	Review66	Review66_R	Review67	Review67_R	Review68	Review68_R	Review69	Review69_R	Review70	Review70_R	Review71	Review71_R	Review72	Review72_R	Review73	Review73_R	Review74	Review74_R	Review75	Review75_R	Review76	Review76_R	Review77	Review77_R	Review78	Review78_R	Review79	Review79_R	Review80	Review80_R	Review81	Review81_R	Review82	Review82_R	Review83	Review83_R	Review84	Review84_R	Review85	Review85_R	Review86	Review86_R	Review87	Review87_R	Review88	Review88_R	Review89	Review89_R	Review90	Review90_R	Review91	Review91_R	Review92	Review92_R	Review93	Review93_R	Review94	Review94_R	Review95	Review95_R	Review96	Review96_R	Review97	Review97_R	Review98	Review98_R	Review99	Review99_R	Review100	Review100_R	Review101	Review101_R	Review102	Review102_R	Review103	Review103_R	Review104	Review104_R	Review105	Review105_R	Review106	Review106_R	Review107	Review107_R	Review108	Review108_R	Review109	Review109_R	Review110	Review110_R	Review111	Review111_R	Review112	Review112_R	Review113	Review113_R	Review114	Review114_R	Review115	Review115_R	Review116	Review116_R	Review117	Review117_R	Review118	Review118_R	Review119	Review119_R	Review120	Review120_R	Review121	Review121_R	Review122	Review122_R	Review123	Review123_R	Review124	Review124_R	Review125	Review125_R	Review126	Review126_R	Review127	Review127_R	Review128	Review128_R	Review129	Review129_R	Review130	Review130_R	Review131	Review131_R	Review132	Review132_R	Review133	Review133_R	Review134	Review134_R	Review135	Review135_R	Review136	Review136_R	Review137	Review137_R	Review138	Review138_R	Review139	Review139_R	Review140	Review140_R	Review141	Review141_R	Review142	Review142_R	Review143	Review143_R	Review144	Review144_R	Review145	Review145_R	Review146	Review146_R	Review147	Review147_R	Review148	Review148_R	Review149	Review149_R	Review150	Review150_R	Review151	Review151_R	Review152	Review152_R	Review153	Review153_R	Review154	Review154_R	Review155	Review155_R	Review156	Review156_R	Review157	Review157_R	Review158	Review158_R	Review159	Review159_R	Review160	Review160_R	Review161	Review161_R	Review162	Review162_R	Review163	Review163_R	Review164	Review164_R	Review165	Review165_R	Review166	Review166_R	Review167	Review167_R	Review168	Review168_R	Review169	Review169_R	Review170	Review170_R	Review171	Review171_R	Review172	Review172_R	Review173	Review173_R	Review174	Review174_R	Review175	Review175_R	Review176	Review176_R	Review177	Review177_R	Review178	Review178_R	Review179	Review179_R	Review180	Review180_R	Review181	Review181_R	Review182	Review182_R	Review183	Review183_R	Review184	Review184_R	Review185	Review185_R	Review186	Review186_R	Review187	Review187_R	Review188	Review188_R	Review189	Review189_R	Review190	Review190_R	Review191	Review191_R	Review192	Review192_R	Review193	Review193_R	Review194	Review194_R	Review195	Review195_R	Review196	Review196_R	Review197	Review197_R	Review198	Review198_R	Review199	Review199_R	Review200	Review200_R	Review201	Review201_R	Review202	Review202_R	Review203	Review203_R	Review204	Review204_R	Review205	Review205_R	Review206	Review206_R	Review207	Review207_R	Review208	Review208_R	Review209	Review209_R	Review210	Review210_R	Review211	Review211_R	Review212	Review212_R	Review213	Review213_R	Review214	Review214_R	Review215	Review215_R	Review216	Review216_R	Review217	Review217_R	Review218	Review218_R	Review219	Review219_R	Review220	Review220_R	Review221	Review221_R	Review222	Review222_R	Review223	Review223_R	Review224	Review224_R	Review225	Review225_R	Review226	Review226_R	Review227	Review227_R	Review228	Review228_R	Review229	Review229_R	Review230	Review230_R	Review231	Review231_R	Review232	Review232_R	Review233	Review233_R	Review234	Review234_R	Review235	Review235_R	Review236	Review236_R	Review237	Review237_R	Review238	Review238_R	Review239	Review239_R	Review240	Review240_R	Review241	Review241_R	Review242	Review242_R	Review243	Review243_R	Review244	Review244_R	Review245	Review245_R	Review246	Review246_R	Review247	Review247_R	Review248	Review248_R	Review249	Review249_R	Review250	Review250_R	Review251	Review251_R	Review252	Review252_R	Review253	Review253_R	Review254	Review254_R	Review255	Review255_R	Review256	Review256_R	Review257	Review257_R	Review258	Review258_R	Review259	Review259_R	Review260	Review260_R	Review261	Review261_R	Review262	Review262_R	Review263	Review263_R	Review264	Review264_R	Review265	Review265_R	Review266	Review266_R	Review267	Review267_R	Review268	Review268_R	Review269	Review269_R	Review270	Review270_R	Review271	Review271_R	Review272	Review272_R	Review273	Review273_R	Review274	Review274_R	Review275	Review275_R	Review276	Review276_R	Review277	Review277_R	Review278	Review278_R	Review279	Review279_R	Review280	Review280_R	Review281	Review281_R	Review282	Review282_R	Review283	Review283_R	Review284	Review284_R	Review285	Review285_R	Review286	Review286_R	Review287	Review287_R	Review288	Review288_R	Review289	Review289_R	Review290	Review290_R	Review291	Review291_R	Review292	Review292_R	Review293	Review293_R	Review294	Review294_R	Review295	Review295_R	Review296	Review296_R	Review297	Review297_R	Review298	Review298_R	Review299	Review299_R	Review300	Review300_R	Review301	Review301_R	Review302	Review302_R	Review303	Review303_R	Review304	Review304_R	Review305	Review305_R	Review306	Review306_R	Review307	Review307_R	Review308	Review308_R	Review309	Review309_R	Review310	Review310_R	Review311	Review311_R	Review312	Review312_R	Review313	Review313_R	Review314	Review314_R	Review315	Review315_R	Review316	Review316_R	Review317	Review317_R	Review318	Review318_R	Review319	Review319_R	Review320	Review320_R	Review321	Review321_R	Review																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													

Figura 5.11: Hoja de cálculo con restaurantes de Miami extraídos de Tripadvisor.

5.3.2. Análisis de preguntas

Tras observar los datos extraídos de Tripadvisor, el usuario podría hacer diferentes preguntas. En este caso se ha decidido que pueden ser interesantes las que se presentan a continuación:

- **¿Me podrías ayudar a buscar restaurantes con comida italiana?** Al igual que se

³Import.io extracción de datos tabulares a partir de sitios web.

puede preguntar por italiana, se puede preguntar por cualquier tipo de comida que se haya extraído de los restaurantes situados en Miami.

- **¿Qué restaurantes vegetarianos hay por menos de 45 dolares?** Similar a la pregunta anterior, con la condición de precio máximo, ya que son dos aspectos que se suelen mirar frecuentemente a la hora de buscar un restaurante, si gusta la comida y el precio.
- **¿Qué opinión hay sobre un restaurante concreto?** Mediante la opinión de otros usuarios extraída del sitio web, el usuario del Chatbot puede deducir si un restaurante es bueno o no.

5.3.3. Definición del DSL

Para este ejemplo se trabajará la definición de las interacciones para las dos primeras cuestiones previamente definidas. En la Figura 5.12 se ve la definición para la primera de las preguntas, dónde la entidad de entrada es la columna *Cuisines* (tipo de cocina) y en la salida se mostrará un mensaje personalizado con el nombre, el tipo de comida, el rango de precio y un link para abrir la ficha en Tripadvisor.

Se ha observado que el resultado que ofrece el chatbot (junto con la previsualización de links de la plataforma Slack) hace que el chatbot al ofrecer la respuesta sea muy verboso. Por esa razón, se ha decidido que se limite a tres el número de respuestas máximas que va a recibir el usuario.

```
"intents": [
  {
    "ID": "restaurantInfoPerCuisine",
    "sourceSheet": "Restaurants",
    "response": {
      "outputColumn": "Name, RangePrice, Cuisines, Name_link",
      "numberOfResponses": 3,
      "customResponseStructure": ["The '%s' restaurant has %s food and it cost between %s. More info via: %s",
        "Name", "Cuisines", "RangePrice", "Name_link"]
    },
    "entities": [{
      "inputColumn": "Cuisines",
      "mask": "LIKE",
      "entityMissingMessage": "Which type of cuisine would you like? For example:",
      "entityNotFoundMessage": "Cuisine style not found, try with:"
    }]
  }
],
```

Figura 5.12: DSL de SheetChat que describe el intent para la búsqueda por tipo de comida de los restaurantes de Tripadvisor.

De igual manera, en la Figura 5.13 se muestra la definición utilizando el DSL de Sheet-Chat donde el usuario hará peticiones respecto a un precio máximo y un tipo de comida concretos, es decir, hay dos entidades participantes a la hora de realizar el filtrado entre

los 400 restaurantes. Como novedad cabe destacar que el valor del campo Mask en lugar de ser LIKE (que sirve para comparar strings), se utiliza el símbolo < para designar que es un valor numérico y que el filtro rechazará los resultados mayores que el precio máximo delimitado por el usuario a la hora de interactuar con el Chatbot, tal y cómo se muestra en el Apartado 5.3.4.

```
{
  "ID": "restaurantsCheaperThanPerCuisine",
  "sourceSheet": "Restaurants",
  "response": {
    "outputColumn": "Name, RangePrice, Cuisines, Name_link",
    "numberOfResponses": 3,
    "customResponseStructure": ["The %s restaurant has %s food and it cost between %s. More info via: %s",
      "Name", "Cuisines", "RangePrice", "Name_link"]
  },
  "entities": [
    {
      "inputColumn": "Cuisines",
      "mask": "LIKE",
      "entityMissingMessage": "Which type of cuisine would you like? For example:",
      "entityNotFoundMessage": "Cuisine style not found, try with:"
    },
    {
      "inputColumn": "MaxPrice",
      "mask": "<",
      "entityMissingMessage": "Set maximum price. For example:",
      "entityNotFoundMessage": "Not found cheaper than that price. Can you set a higher price as: "
    }
  ]
}
```

Figura 5.13: DSL de SheetChat que describe el intent para la búsqueda por precio máximo (resaltado en azul) y tipo de comida de los restaurantes de Tripadvisor.

5.3.4. Ejemplo de uso del chatbot

Tras el diseño del Chatbot mediante el DSL de SheetChat a continuación se presenta cómo sería la experiencia de uso de este chatbot por parte de un usuario.

En la Figura 5.14 el usuario en primer lugar saluda al Chatbot y este le ofrece alguna de sus funcionalidades por si acaso el usuario no se acuerda de qué trata este chatbot o qué funcionalidades tenía. Posteriormente se dispone a preguntar por restaurantes italianos. El chatbot le responde con tres resultados, el número máximo de resultados que se indicó en el diseño del chatbot (ver Apartado 5.3.3).

En la interacción de la Figura 5.15 se observa que al usuario las recomendaciones de los restaurantes italianos le ha resultado cara. Por lo tanto decide preguntar por restaurantes con precio menor a 15 dolares. El chatbot ha detectado que el usuario está preguntando por restaurantes con un precio máximo y un tipo de cocina determinado. El chatbot restringe las respuestas en base a ese nuevo filtro y muestra el único restaurante que cumple esas características. En caso de que no hubiese ningún restaurante que cumpliese ambos criterios preguntaría por las entidades nuevamente.

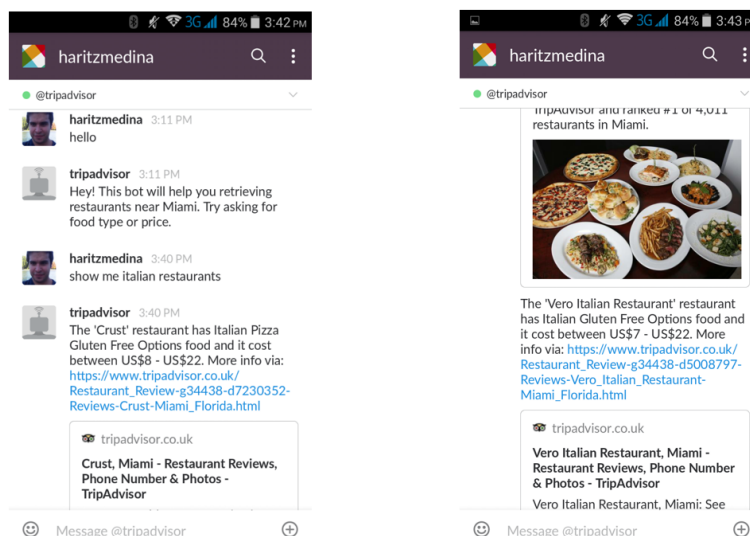


Figura 5.14: Interacción entre el usuario y el chatbot que recomienda restaurantes en base a un tipo de cocina.

De esta manera, mediante el uso de un chatbot, el usuario evita tener que aplicar filtros en el sitio web de Tripadvisor que en una configuración móvil suele resultar en líneas generales bastante farragoso.

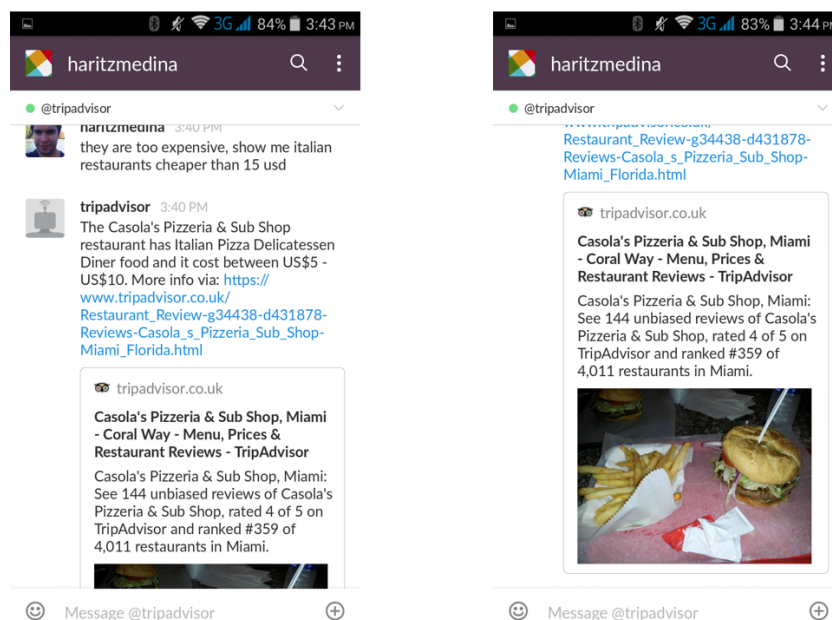


Figura 5.15: El bot de Tripadvisor recomienda restaurantes italianos con precio menor a 15 dolares por petición del usuario.

6. CAPÍTULO

Conclusiones

Anexos

A. ANEXO

import.io: Extracción de datos tabulares a partir de la web

B. ANEXO

wit.ai: Procesamiento del lenguaje natural orientado a bots conversacionales

Bibliografía

- [Dans, 2016] Dans, E. (2016). El bot como interfaz preferida. <https://www.enriquedans.com/2016/06/el-bot-como-interfaz-preferida.html>. Accedido: 2016-09-11.
- [Ferrara et al., 2016] Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. (2016). The rise of social bots. *Communications of the ACM*, 59(7).
- [Montag et al., 2015] Montag, C., Błaszczewicz, K., Saryiska, R., Lachmann, B., Andone, I., Trendafilov, B., Eibes, M., and Markowetz, A. (2015). Smartphone usage in the 21st century: who is active on WhatsApp? *BMC research notes*, 8:331.
- [Wagner, 2016] Wagner, K. (2016). Bots, explained. <http://www.recode.net/2016/4/11/11586022/what-are-bots>. Accedido: 2016-09-11.
- [Wang et al., 2016] Wang, D., Xiang, Z., and Fesenmaier, D. R. (2016). Smartphone Use in Everyday Life and Travel. *Journal of Travel Research*, 55(1):52–63.
- [Wright, 2016] Wright, D. (2016). Are digital assistants and chat bots the same thing? <https://www.linkedin.com/pulse/digital-assistants-chat-bots-same-thing-david-wright>. Accedido: 2016-09-13.

Agradecimientos
