

AndyBot

Sistema de Desarrollo

para crear

Agentes Inteligentes

(v: 7.00)

Introducción

En un mundo en donde la interfase ya impuesta con una computadora, es un teclado¹ y mouse², surge la pregunta obligada y minimalista ¿Cuánto esfuerzo se requiere para aprender a usar esto para tareas simples como anotar algo en la agenda para no olvidarlo mañana, imprimir este mismo documento, contestar el mail de Pablo con “voy al asado”.

Está demostrado que aunque haya sistemas usados masivamente por toda la gente, éstos no son óptimos, pues en este caso nosotros los humanos, luego de millones de años de optimización genética adaptativa (*Darwin mediante*) nos comunicaríamos con clicks y movimientos de la mano en un aire “virtual” (*como algunas películas futuristas*). En realidad esto no es hoy así y tampoco parece que lo vaya a ser en un futuro cercano, por más que la tecnología ya está bajo prueba (*surface, etc.*).

En cambio por lo general nosotros, los humanos, usamos cosas más robustas como el lenguaje, transmitiendo ideas mediante secuencias de señales físicas, cosa muy adoptada por una gran mayoría de los seres vivos. Si bien hay métodos muy sofisticados de comunicación natural como la visual (*seres del abismo con fotoluminiscencia, o las Jibias, unos cefalópodos que son como minipantallas de video*), el mecanismo escogido por la naturaleza terrestre es convertir las ideas o “conceptos” en algo comunicable como sonidos audibles y/o gestos visibles, croar, piar, rugir, gruñir, rebuznar y por último el humano inauguró el hablar. En nuestro caso esas señales forman un lenguaje sofisticado, formado por unidades o segmentos fonéticos unidos a los que mayormente llamamos palabras, las cuales también tienen codificación en imágenes como el texto escrito (ideogramas y grafemas) y simultáneamente es desglosado como sonidos producidos por nuestro aparato fonador (lengua, dientes, labios, garganta, nariz conjuntamente con nuestra voz).

Para nosotros esto, la conversación es el modo de comunicación más natural y sencillo.

¿Por que ir contra la corriente?

¿No es más fácil preguntarle a un operador humano si me queda dinero en la cuenta de ahorro para pagar el gas de \$200 este mes? Si optamos por hacer esto hoy, por **callcenters** de asistencia que ofrecen ciertas entidades; deberemos de armarnos de paciencia y estar dispuestos a escuchar un pesado y frecuentemente complicado sistema de opciones con los dígitos del teléfono, *digite 1 para cuentas, 2 para corporativo.. etc.* Llamado IVR (*Interactive Voice Response*) Luego si con suerte estamos en el horario correcto, el día adecuado, habrá alguna opción generalmente escondida que nos conecta con un operador que *‘casi siempre está ocupado con alguien más importante que Ud’*. Si luego de esos largos minutos de jugar en el teclado digitando opciones, y escuchando cada una atentamente, logramos ingresar las claves que nos piden cambiar todos los meses y logramos finalmente hablar con alguien, probablemente ya nos hayamos olvidados para que llamamos al callcenter y probablemente nunca mas lo hagamos, menos para saber algo tan poco importante como lo que nos preocupa, perdiendo todos esos minutos (son contar el disgusto, si es que no anduvo). No contamos en esto el costo de la llamada, frecuentemente a cargo importunado usuario.

Por el otro lado una opción actual, lógicamente más usada es entrar al home-banking por Internet (*media docena de clicks, varias claves, tarjetas de coordenadas con claves tipo batalla naval o aparatos electrónicos tipo llavero con secuencias de números especiales*), si luego ver una pantalla nueva (*pues cambia frecuentemente*), y luego de leer todo, estimar que lo que queremos está bajo una opc.A, entrar luego en sub-opc.2, luego ir a opc.K, poner el código y tipo de cuenta, luego click en sub-opc.4, y luego presionar OK y esperar el ticket!
¿es necesaria tanta complicación, para algo tan simple?

Hoy le damos una buena noticia, esto está a punto de cambiar...!

¹ Sistema de ingreso de texto gráfico-óptico-manual, estimo máquina de escribir.

² Dispositivos complejos llamados pointing-devices, además de lápiz, touchscreen, etc.

Un sistema de Diálogo “Natural”

El sistema **AndyBot**, es un mecanismo orientado a poder generar fácilmente un sistema cuya exteriorización es la de un agente inteligente, capaz de conversar con una persona en forma natural, racional y lógica, respondiendo a preguntas, siguiendo conversaciones, presentando declaraciones, propuestas, sugerencias y mucho más; en definitiva entablando un diálogo, tal como lo haría un humano real. La finalidad y transcurso del diálogo es dependiente de cada necesidad. Esto es tan variado como lo son las necesidades de la gente y las empresas.

El sistema permite crear rápidamente y con simplicidad, servicios de conversación en castellano o inglés (*u otro idioma habilitado*), resultando diálogo coloquial para su uso en aplicaciones como: promociones, ventas, encuestas, asesoramiento, asistencia técnica, entretenimiento, consultas y transacciones en sitios web, entre tantos otros.

El sistema, usado como base de diálogo, podrá contestar una enorme cantidad de información pues posee la posibilidad de acceder a mucha información para evacuar dudas y preguntas, tanto en bases de datos dedicadas como en la web misma.

El sistema **AndyBot**, fue creado por especialistas en inteligencia artificial y lingüística computacional, con muchos años de experiencia como investigadores en la UBA y el Conicet. Se han volcado en él los últimos avances de sistemas de diálogo, trabajos científicos recientes publicados por los autores e incluso ciertas técnicas inéditas aún no publicadas en la comunidad científica. El sistema implementa un selecto conjunto de lo mejor en inteligencia artificial y lingüística computacional de la actualidad, aplicado al diálogo hombre-máquina.

¿Qué se puede hacer AndyBot?

Generar un sistema de diálogo con un Agente Inteligente el cual posee las siguientes características destacadas:

- Reconoce Lenguaje Español y/o Inglés
- Diccionario Morfológico con Reconocimiento de >2.5 millones de palabras
- Corrección de Errores de ingreso (ortografía/omisión/teclado/etc.)
- Reconocimiento de palabras y grupos de palabras
 - Semántico y Morfológico (verbos conjugados y palabras flexionadas)
 - Gramatical de partes de la Oración (sintaxis)
 - Palabras Inexistentes (Parasintéticas)
 - Unidades Internacionales (SI, MKS, CGS: es, en)
 - Cifras (numéricas, coloquiales, romanos, hexa, etc.)
 - Fechas y Tiempos (Calendario y Horarios)
 - Locuciones (es:4500, en:1200)
 - Entidades y Nombres Propios (es:18.500, en:8.500)
 - Reconocimiento Fonético de palabras mal escritas (similitud)
 - Distinción entre palabras locales, extranjeras e impronunciables (basura)
- Facilidad de extracción de información
- 65 Funciones Lingüísticas.
- Cadenas de Razonamiento Temático Multiconectado.
- Mecanismo “casi humano” para seguimiento del hilo de una conversación.
- Generación de Respuestas basadas en datos.
- Capacidad de Flexión (en la generación del diálogo de respuesta)
- Extracción de información de los datos volcados por el usuario.
- Persistencia de Información en un diálogo para con cada usuario.
- Velocidad de Análisis y Respuesta (>20.000 cláusulas por segundo)
- Posibilidad de sostener numerosas conversaciones simultáneas (>500)

Pasos Generales para Crear un Sistema de Diálogo

Un sistema implementado con **AndyBot** puede contestar dentro de un amplio rango de cosas,

Constucción de un Agente Inteligente

¿Cómo se hace esto?

Este es un simple paso a paso: 1-2-3-4,

Ejemplo:

1. Definir cual va a ser el ámbito de aplicación de los diálogos
 - a. Si va a contestar cosas de una base de datos
 - b. Si va a tener respuestas coloquiales y personales
 - c. Si va a ser "creativo" en las respuestas (IA).
2. Modelar y definir los diálogos deseados:
 - a. Proveemos un ambiente de desarrollo on-line con:
 - i. Compilador **AndyScript**
 - ii. Editor de Texto con Analizador de Sintaxis
 - iii. Debugger (pruebas en vivo on-line)
 - iv. Sistema de Control del Servicio
 - v. Actualización Hot-Swap (sin detener el servicio)
 - b. No se requiere conocimientos de programación
 - c. Si es necesario, se puede programar (**AndyScript, C# / .NET**)
3. Definir mecanismos de conexión: MSN / web / SMS / etc.
 - a. Para MSN se requiere un contrato con MS. para mas de 500 usuarios.
 - b. SMS requiere de contratos y V^oB^o de los operadores (**Celcos**)
 - c. Web y otros chat, no tienen limitaciones de numero de usuarios
4. Contratar el servicio con un esquema comercial SAS
 - a. Licencia de puesta en marcha (Setup-fee)
 - b. Costo mensual del servicio (fijo + variable)
 - c. Actualizaciones periódicas, revisión e inclusión de nuevos diálogos, etc.

Sin entrar en detalles complejos, el sistema se crea y programa con un mecanismo sencillo pero riguroso, dentro del cual se deben seguir ciertas reglas para poder llegar a un puerto seguro y tiempos acorde a lo deseado. Describimos un poco este mecanismo, si bien el orden es sugerido, no es impuesto en lo absoluto.

Diseño de un Agente

Procedimiento General

- Se determina el ámbito de **cosas que debe saber y responder**, por ejemplo un help-desk de un banco o tarjeta de crédito, un sistema de sugerencias, un sitio web, etc.
- Se define un equipo de gente capaz de conocer la lógica de negocios del agente, éste grupo será llamado **Modelador de Diálogo** Será el encargado de crear la lógica y eventualmente solicitar la asistencia de programación a la empresa nuestra, conforme a lo que se desea realizar con el agente.
- Los pass son sencillos pero se definen fácilmente
- Primero se definen los **rubros** en que se puede incurrir, durante el diálogo.
- Dentro de cada **rubro**, se arman las típicas '*conversaciones figuradas*', suponiendo todas las inquietudes del cliente. "si pregunta '*esto*' responder '*esto-otro*'

- Para esto se encadenan preguntas y respuestas, como en un simple guión o novela. (esto se verá, en la sección de **Diagramación y Modelación de Diálogo**)

Capacitación

- Se brindará Capacitación al equipo Moderador de Diálogo para el uso básico del lenguaje de especificación de diálogo “**AndyScript**”
- Opcionalmente se podrá brindar capacitación más sofisticada para programación a un equipo IT de la empresa cuando haya políticas tipo in-house (por ejemplo un banco)

Resultado Esperable

1. **Disponibilidad**
El Agente siempre estará on-line (durante el tiempo que se desee y con la agenda requerida)
2. **Preguntas y Asistencia**
El Agente Contestará por lo general en pocas fracciones de segundo, con la información precisa o con una repregunta solicitando aclaraciones. En muchos casos, el Agente podrá re-consultar cosas específicas y datos al usuario para una respuesta mas precisa, siguiendo el hilo de las conversaciones, sin limitaciones.
3. **Encuestas**
Se puede hacer que el usuario/cliente responda una encuesta para actualizar información en la base de datos.
4. **Llenado Selectivo de Fichas**
El Agente puede obrar de filtro inicial para una consulta de requisitos a un potencial cliente, llenando una ficha y conforme a las respuestas recibidas, formular las preguntas necesarias; a la vez de estar brindando las explicaciones que el usuario requiera en cada momento.
Este es el caso de un Agente consultor para telemarketing, o a un sitio web de venta de productos especiales en donde el cliente debe reunir una determinada cantidad de requisitos para aplicar al mismo o a que se le ofrezca determinado producto, por ejemplo un crédito para un auto, o una oferta laboral determinada, la cual depende de la capacitación del mismo.
5. **Información**
Todo diálogo acaecido, queda guardado en una base de datos junto con los datos pormenorizados y recabados durante éste diálogo.
Es posible extraer esta información accediendo a una base de datos la cual fue enriquecida durante los sucesivos diálogos. El sistema determinará cual es la información importante a obtener de cada usuario, para esto de cada cosa que el usuario dice o responde, (dependiendo de la posición dentro del diálogo), se obtienen registros que pueden guardarse en una base de datos extra.

Derivación Programada

Es posible que un diálogo llegue a una etapa, en donde es necesario que la restante parte sea hecha por una persona, en este caso se le deriva la conversación completa y se le indica realizarle el resto de las preguntas a un operador humano (*por ejemplo para concretar una cita, hablar con un supervisor, u otra acción deseada*). → El operador, recibirá el detalle de la conversación anterior, junto a todos los datos necesarios y recabados del cliente para seguir con su trabajo con el cliente.

Comunicación y Capacidades Especiales

- Vías de comunicación: CHAT-Agent con MSN y/o Yahoo-Chat (Skype y Google-Talk en desarrollo), vías SMS y vía website (directo desde el sitio web)
- El agente puede recibir mails y contestarlos, avisando de estos y otros eventos al usuario en determinados momentos (todo esto es configurable)
- Durante una conversación por MSN, el agente puede recibir archivos, fotos y documentos (por las clásicas transferencias peer-to-peer) y además posee inteligencia para determinar si es viable o aceptable la recepción o si tiene un virus. Esto es configurable en cuanto a tipo de archivos, tamaño, totales de transferencia por usuario (quote) y quienes pueden o no recibir archivos y de que tipo.

Generalidades

- El sistema corre como SAS (Software As a Service) en un servidor de alta performance en nuestro **Datacenter**, si el requerimiento de tráfico es muy alto, se puede implementar un servidor dedicado para el cliente. Si requiere de alta disponibilidad, deberá haber varios servidores individuales dedicados. (consultar costos)
- Capacidad de carga:
 1. hasta 10 respuestas por segundo por servidor
 2. > 2.5 Millones de consultas por mes. con hasta 100 usuarios simultáneos, respondiendo a cada uno en fracciones de segundo (promedio), sin contar la demoras de la web y de los sistemas de chat (MSN/Yahoo/GoogleTalk).
- Cantidad de usuarios diferentes: por web es ilimitada; para chat por MSN se pueden tener hasta 500 usuarios registrados simultáneamente, para mas cantidad se debe contratar una cuenta corporativa con Microsoft (consultar costos)

Diagramación y Modelación del Diálogo

El mecanismo de diagramación del diálogo requiere poco esfuerzo por parte del cliente-programador, quien no necesita saber programar en ningún lenguaje en especial, solamente saber describir un diálogo, similar a como luce en una novela o cuento, con el diálogo de ida y vuelta. La potencia y expresividad del sistema es enorme pudiendo utilizar una gran cantidad de funciones muy útiles a título léxico para crear ricos mecanismos para el control del flujo en el diálogo con mucha facilidad.

Se proporciona una gramática básica y ampliable como un pseudo-lenguaje de descripción del diálogo mismo de interfase directa a programación en lenguaje C# y VB en forma nativa.

Por lo tanto, si se poseen capacidades de programación en lenguajes simples como el propio llamado **AndyScript** y se desean ampliar con C#, VB o J# u o cualquier lenguaje que compile bajo **.NET** o **Mono**; se podrá hacer uso de funciones específicas orientadas a la lengua y especialmente diseñadas para facilitar el modelado del diálogo mismo.

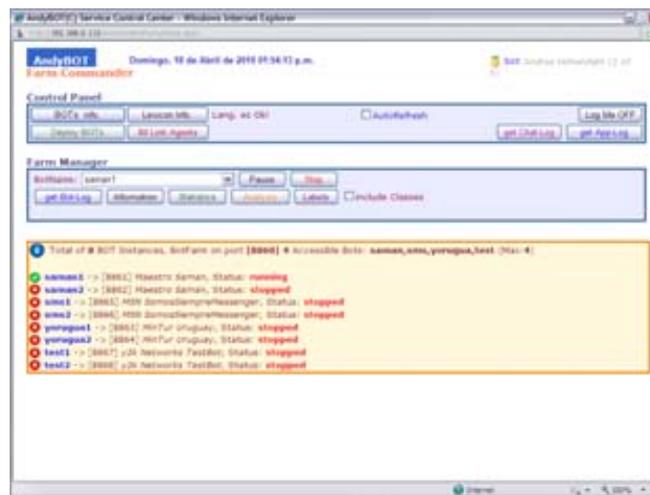
Todo esto, crea una herramienta muy flexible que permitirá la programación de tareas con manejo de lengua y texto, en unas pocas líneas de código muy entendibles e intuitivas. Estas tareas de modelado de lengua, a veces parecen simples, son habitualmente muy complejas e ineficientes con programación convencional. En cambio el sistema presentado está orientado en forma directa al procesamiento de Texto Natural brindando mecanismos intuitivos, eficientes y sencillos de usar.

Pantallas del Sistema

Se mostrarán a título de ejemplo, algunas de las pantallas del sistema actual, utilizadas ya desde 2009 por un equipo de programadores y periodistas para modelar el diálogo de varios agentes comerciales implementados.

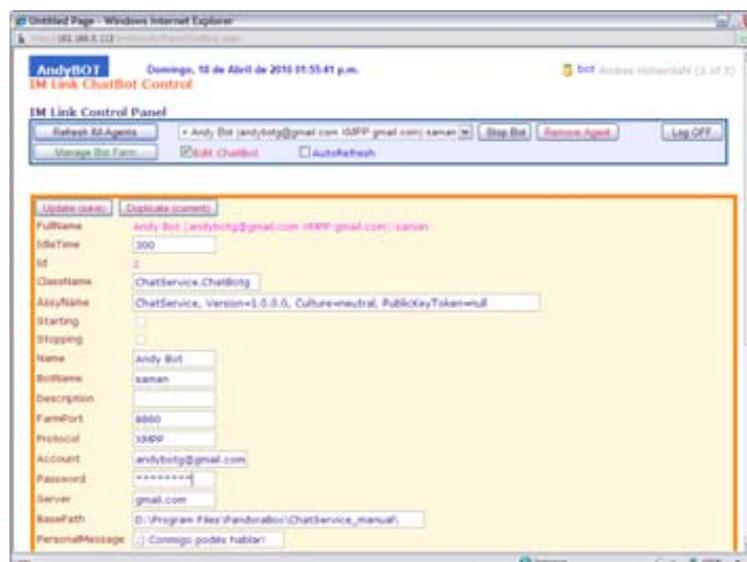
Sistema de Gestión de Granja de BOT's

Este sistema es responsable de dar de alta, controlar su ejecución y estadísticas de una granja de varios **bots**. Nótese que para cada **bot** se usan 2 versiones, puesto que el sistema se ha diseñado para Alta Disponibilidad, en otras palabras si un **bot** está en uso y se debe actualizar, éste nunca se desconectará de sus usuarios, inclusive no perderá conversaciones en curso, simplemente se iniciará el nuevo **bot**, se le pasan todas las conversaciones en curso al nuevo y recién luego de la última respuesta, se apaga el anterior.

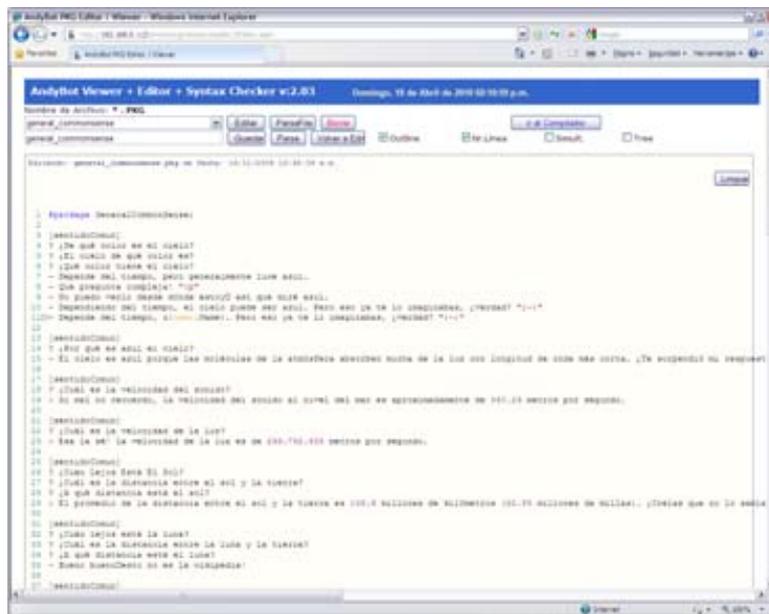


Control de Vínculos IM

Esta pantalla muestra el control detallado sobre cada vínculo de IM, en donde según corresponda, se podrá tener control detallado de cada vínculo con un agente de Chat (IM) en este caso se muestra un agente usando protocolo tipo Jabber (XMPP) para conectarse con la red IM de chat de **Google** llamada **Gtalk**.

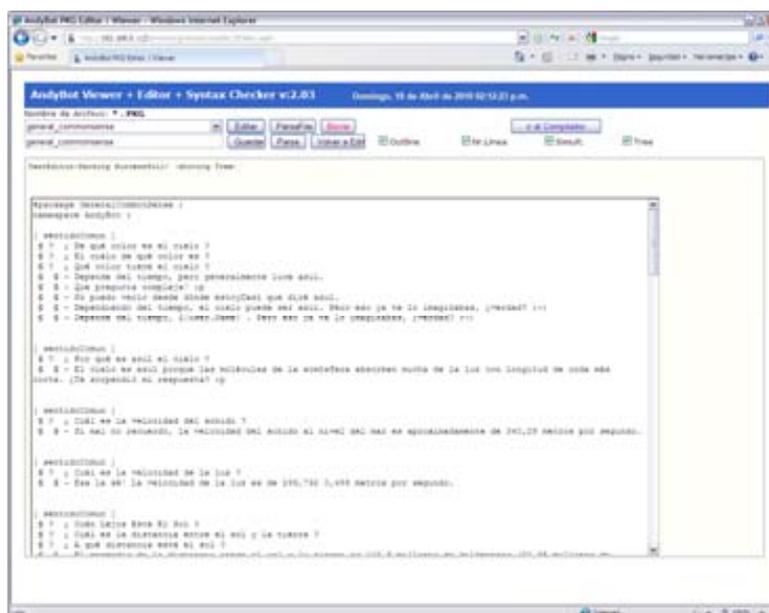


Editor: Posee además soporte de control de sintaxis, con compilación temprana para detección de errores (*parsing*). En caso de errores, el sistema indica claramente y realiza un salto directo hacia el sitio exacto donde se determinó que hubo un error de sintaxis (*línea y columna*) con explicación detallada del mismo, en la cabecera de la pantalla.



Visualización de Flujo de Diálogo

El sistema **editor** posee un modo de visualización llamado **“Tree”** (árbol) que muestra las relaciones jerárquicas de respuestas-preguntas para el seguimiento de diálogo, de modo de poder seguir claramente las relaciones al diseñar el diálogo esperado.



Este editor, conjuntamente con el compilador y administrador, corren vía web, en un cómodo y seguro ambiente de edición remota.

Flujo de Diálogo Básico, un ejemplo

El usuario dice algo y el sistema responde conforme a lo que dijo. En otras palabras si el usuario dice:

hola!

Se podría responder coloquialmente:

Cómo estás?

La sintaxis que proponemos es la siguiente: usamos una interrogación en todo lugar en donde se está preguntando que dijo el usuario y un guión antepuesto a la respuesta correspondiente. Ejemplo:

? Hola!
- Cómo estás?

Tal parece que esto parece que solamente contestará a Hola!, y es cierto! Lo cual no es muy cómodo, locuaz ni abarcativo. Además contestaría siempre lo mismo. Por lo cual se previno la posibilidad de que tenga varias respuestas posibles, simplemente agregando respuestas a continuación, por ejemplo:

? Hola!
- Cómo estás?
- Que hacés, como anda eso?
- Me alegra charlar contigo de nuevo!

Pero esto solamente contestará a una sola pregunta (que en adelante llamaremos patrón de coincidencia) Veamos como podemos ampliar la posibilidad de análisis, simplemente agregando otra forma de coincidencia, por ejemplo:

? Hola!
? Que tal?
- Cómo estás?
- Que haces, como andas hoy?
- Me alegra charlar contigo de nuevo!

Esta es la base mínima del sistema, pero para poder entender como funciona en profundidad debemos ver algunas cosas paso a paso y cuando se haya leído todo completamente esta guía, se verá la verdadera potencia del sistema. Tal vez haya que leerlo varias veces.

En las próximas secciones repasaremos las partes constitutivas del sistema, para entender perfectamente que hace y como funciona, en total detalle a fin de poder aprovechar al máximo la capacidad de generar un sistema de diálogo natural y flexible.

Cadena de Razonamiento

Para explicar como funciona el **flujo del diálogo** desde que el usuario se expide hasta recibir una respuesta, detallaremos el camino y las decisiones realizadas, en forma simplificada.

El sistema arranca con un Diccionario, el cual reconoce las palabras y/o 'cosas' escritas por el usuario, este diccionario posee una gran responsabilidad, pues es quien debe identificar las palabras y símbolos, corregir eventuales errores de ortografía, reconocer abreviaturas, locuciones, números dígitos numéricos, unidades y estimar palabras inexistentes, entre otras cosas.

Luego que la frase de ingreso es *tokenizada* = (troquelada o cortada en trozos: *tokens*) ésta se somete a una serie de comprobaciones, para ver si coincide con un patrón de la conversación que deseamos capturar para dar la respuesta pertinente.

Ahora bien, el diálogo es algo dinámico y no se puede responder siempre lo mismo, y la respuesta tendrá valor conforme haya una pregunta anterior que espere este tipo de respuesta.

Este mecanismo de desarrollo de diálogo, ha sido refinado y modelado con una fuerte reminiscencia a la manera de pensar de las personas, utilizando memoria, capacidad de retomar diálogos anteriores, etc. Iremos viendo esto en profundidad a medida que avancemos en la explicación del sistema.

Ahora hablaremos del sistema de reconocimiento de palabras.

Diccionario Inteligente con Reconocimiento Lingüístico

El sistema **AndyBOT** viene incorporado con funciones específicas que facilitan la captura de información a partir de un texto escrito, pudiendo separar e identificar palabras, números, siglas y demás simbologías. El resultado es una secuencia de palabras, abreviaturas, emoticones y siglas que corresponden al diálogo escrito. Esta secuencia de palabras reconocidas, luego se pasa por módulos de reconocimiento de gramática para aislar partes de la oración y construcciones para identificar los conceptos y otros patrones de búsqueda.

Lematizador Morfosintáctico

El sistema posee la posibilidad de reconocer un número enorme de palabras teniendo el conocimiento de cómo están formadas morfológicamente las palabras, a fin de poder interpretar la conjugación o flexión (cambio morfológico que hayan sufrido) Reconoce género (masculino / femenino), número, (plural / singular), diminutivos, peyorativos, aumentativos, conjugación de los verbos.

Una característica sobresaliente es el poder comprender palabras parasintéticas o sea palabras que se inventan y no existen en diccionario alguno; pero son entendibles por un humano como una derivación de algo conocido, con un sentido dado, como por ejemplo: *perruchitas*, reconocido como '*posible*' *diminutivo femenino plural* de *perro*; en otras palabras el sistema la reconoce como *plural, diminutivo de perro* asignándole un costo por la no-existencia en diccionario, para dejar sentado esto.

Para lograr esto, el sistema posee reconoce varios centenares de prefijos y sufijos diferentes, algunos combinables con otros, conformando palabras mucho más complejas, usuales en medicina, física, ciencias y generalmente en diálogo informal.

A esto se le llama reconocimiento de palabras parasintéticas, lo cual reiteramos, es la generación de palabras conforme se desee expresar algo, usando partes con significados conocidos y ensamblándolas en forma intuitiva, lo cual es más habitual de lo que parece.

Corrección de Errores

Debido a que el texto de diálogo abierto suele contener numerosos errores de escritura, el sistema posee capacidades específicas diseñadas para tratar de sobrellevar estos inconvenientes del mejor modo posible.

Algunos errores comunes son sistemáticos pues están asociados a fallas de tipeo por proximidad (tocar la letra más próxima, en un teclado tipo '**qwerty**', o no pulsar el número adecuado de veces en un teclado de un celular) y otros ya son clasificados como errores por no saber escribir bien o desconocer el idioma, o tener un teclado no configurado para acentos y marcas diacríticas, como acentuación, eñe y diéresis.

Otros errores menos graves son la falta u omisión de mayúsculas y minúsculas en los nombres propios, abreviaturas, principios de oración y siglas, falta o colocación errónea u omisión de marcas diacríticas: acentos, diéresis y cremas.

Para subsanar esto el sistema posee un poderoso sistema para corrección de errores inherentes de ingreso, el cual permite no solo corregir sino provee una lista con un número predeterminado de las palabras más probables por parte del usuario humano, escogiendo la más adecuada según se instruya en el patrón de coincidencias.

Corrección de Palabras combinadas con números y letras sueltas

En la jerga del chat, han surgido innumerables palabras nuevas que son interpretables fácilmente si se las pronuncia de algún modo “en la cabeza” con los nombres de sus partes constitutivas (letras sueltas y números) como ser **salu2** es como decir **salu+dos** o sea **saludos**. Esta es una característica deseable de ser interpretada, pues el sistema lo realiza transparentemente (controlado por un parámetro o *flag*), La interpretación final dependerá si la palabra resultante es “viable” o no. Ejemplos: sali3 = salitres, h2 = hachados, t2 = tedos, entre otras.

Capacidad de Interpretación casi Humana

Este sistema, por tratar de imitar la captación de conceptos y palabras, lo hemos dotado de una capacidad hasta ahora muy inherente al ser humano, que consiste en reconocer palabras “parecidas” más no solo por ortografía y contexto sino también por como “suenan”.

Si por ejemplo si se activa el reconocimiento fonético y se le introduce la palabra VAYEMA, el sistema entenderá esta palabra inexistente como si hubiese puesto BALLENA, solo que penalizará un poco la coincidencia, en caso de estar dentro de algún patrón de búsqueda o resultado. Esto no lo puede hacer cualquier sistema del estado del arte actual!

Esta capacidad de reconocimiento fonética es extremadamente útil pues podrá reconocer palabras de jerga como KASA cuando se quiere decir CASA, si se pone KE entenderá QUE, La lista de cosas raras frecuentemente mal escritas es muy larga, ningún sistema realizable y práctico podría tener todas esas palabras preprogramadas, por esto es que hicimos un desarrollo con inteligencia artificial para un trato mas humanizado y flexible del diálogo.

Diccionario-Lematizador

Este diccionario es un módulo externo, corriendo como un servicio de Windows2000/3/8 o como un daemon (en **linux**) el cual se comunica con el agente, por medio de mecanismos eficientes y directos de objetos remotos “remoting” y/o web-services. Esta solución ha sido escogida como conveniente de modo que haya un solo módulo de Léxico logrando optimizar velocidad y economizar memoria, el proceso es secuencial: llega la frase, se analiza y se devuelve el análisis completo, luego llega otra y así sucesivamente. La velocidad de lematización específica, es de más de 5000 palabras por segundo, dependiendo del tamaño del diccionario y la velocidad del procesador. El consumo de RAM es de entre 40 y 90 megabytes, fluctuando conforme caprichos del garbage collector de .NET.

El diccionario, si bien es fuertemente algorítmico, es ampliable con una lista de palabras del usuario, indicando las etiquetas necesarias en cada una.

El sistema analiza las palabras y responde utilizando un sistema de etiquetas basadas en el estándar de EAGLES 2.0 ampliadas especialmente para aumentar la potencia de análisis, extendiendo funcionalmente y semánticamente las mismas.

Actualmente el diccionario reconoce más de 2.5 millones de palabras exactas (*versión en español*) y 1.2 millones en su versión en inglés. Siendo enormemente tolerante cuando se activa la corrección de errores y el modo fonético; reconociendo texto inventado (*por parasíntesis*) o que no está explícitamente en los diccionarios pero que “suena” razonable a un humano, como ser: *riquísimo, recontrabueno, chiquititita, inimaginables, neumotoráccico*.

Por eso decimos que nuestro sistema es capaz de “humanizar” sustancialmente el diálogo entre un humano y una máquina, entendiendo las palabras por su constitución morfológica y fonética más probable, tal como haría un humano promedio, con un alto grado de cultura.

Casos Especiales

Debido a que el usuario puede ingresar cualquier texto imaginable y para evitar que el sistema realice un esfuerzo innecesario y exagerado para corregir el texto ingresado y buscarlo en un diccionario sin necesidad; se han dispuesto determinadas limitaciones muy lógicas, a saber:

Máxima longitud de palabras (reconocibles como palabra)

34 caracteres (es dependiente del diccionario interno)

Máxima cantidad de número (dígitos) repetidos e iguales = **19 en total**

Máxima cantidad de veces en que un carácter-letra (precedido por otro igual) se repite = **3**
(todo carácter que se repita más de 3 veces será limitado a 3, sin arrojar error,)

Ejemplo:

Si el usuario ingresa

hhhhhhhhhhhhhhoooooooooooooooooiiiiiiiiiiiaaaaaaaaaaaaaa

El diccionario-lematizador lo transformará antes de analizarlo, del siguiente modo:

hhhooollaaa

→ hola_I~0,8

→ X~0,21

Esto significa que la palabra inicial fue limitada a 3 repeticiones de cada letra única como máximo. La resultante es evaluada como

hola I (*Interjección, corregida con una probabilidad de 0.8*)

Además de la propia palabra: **hhhooollaaa** como una palabra desconocida, de rasgos fonéticos foráneos "X" con probabilidad de ser cierta de 0.21 (*menor que la de hola pues no es una palabra no pronunciable como española*)

IMPORTANTE: Debido a este motivo de conversión, no tiene sentido en un patrón de coincidencia especificar mas de 3 letras iguales seguidas (pues jamás ocurrirá)

? Hoooooola!
- Cómo estás?

En este ejemplo nunca será reconocido, pues Jamás se obtendrá del lematizador un texto capaz de coincidir con el patrón, pues limitará a 3 letras iguales repetidas como máximo.

Locuciones y Entidades Nombradas

El sistema permite reconocer grupos de palabras cuya función es única, basado en un extenso diccionario, estos grupos se dividen en dos grandes grupos: Locuciones y entidades nombradas llamadas NE (Named Entites)

Tal es el caso grupos de palabras en una oración como ésta:

El tren pasa **a cada rato**

Esto funciona como una única función gramatical y es un **Adverbio Modal-Temporal** y en última instancia funciona como una interjección usada así:

¡a cada rato!

Si tratásemos de detectar con un patrón de búsqueda cosas como un componente de este grupo por ejemplo la palabra "rato" cometeríamos un error grave de reconocimiento pues no se

puede n separar sin alterar su sentido. Por esto es que el lematizador los trata como si fuesen una sola entidad llamada locución y las etiqueta como adverbio y/o locución.

a_cada_rato:
RG (Adverbio General)
I (Interjección)

Hay que tener cuidado puesto que este sistema cambia la manera de funcionamiento de los patrones, los cuales para hallar esta interjección el particular debiera de especificarse lo siguiente:

? * “a cada rato”
- Parece que las cosas ocurren seguido!

Esta es la manera de capturar este tipo de palabras
Otra manera sería usando el patrón gramatical:

? * {&RG}
- Parece que las cosas ocurren seguido!

Pero cuidado al modelar la respuesta puesto que esto “atrapará” toda frase vacía o no terminada con un adverbio general “RG”.

Mecanismos de Control del Lematizador

El Lematizador reconoce secuencias de caracteres y hace el mejor esfuerzo para detectar los componentes léxicos

Los mecanismos son muy extensos y selectivos pero su utilización para los fines de un diálogo libre en un idioma se configuran por defecto y son los siguientes:

Uso de Diccionarios

UseStandardDict
UseExternalDict

Sección de Lematizado

CheckPrefix | CheckSuffix | CheckRoot | CheckInfix

Corrección de Ortografía

UseSpellCorrection

Reconocimiento

OnlyValid

Etiquetado de palabras desconocidas

| UseHeuristics
| UseMorphoGuess
| UseSuffixGuess
| UseFreelingGuess

Reconocimiento Numérico

| FloatSciNumbers | MultiWordNumbers | WordNumbers | ColloquialNums

Reconocimiento de Moneda, Unidades y Fechas

| RecognizeMoney
| RecognizeUnits
| RecognizeDates

Reconocimiento de Locuciones

| RecognLocutions | MergeLocutions

Reconstrucción Fonética

| PhoneticMatch

Abreviaturas

| RecognizeAbreviated | ExpandAbreviated

Algunas aclaraciones, el diccionario cuenta con una sección de expansión de abreviaturas, en donde se pone **Dr.** y el sistema lo reconocerá como una abreviatura. Como está especificada la opción **ExpandAbreviated**, el sistema expandirá la misma, asignando la raíz de la abreviatura a la etiqueta lingüística. Veamos un ejemplo.

Ejemplo:

Dr. Jorge	→	
Dr.		Yt<*titulo>
Jorge		NPMS

Aquí se observa que, de no haber reconocimiento de la abreviatura, sería una palabra desconocida **Dr** (con su consecuente expansión de posibles palabras por parte del corrector ortográfico: Ej: **Dar, D**, etc.) seguidas de un punto decimal que actúa como separador de oración, cosa no muy útil a la hora de reconocer una persona con un título puesto como abreviatura.

Lenguaje de Definición del Diálogo

Aquí presentaremos un adelanto simple de cómo se especifica un diálogo, es muy similar a un libreto, un guión o una novela, en donde se usan guiones y signos de pregunta para indicar a uno y otro actor en el diálogo mismo, poseyendo mecanismos específicos para poder encadenar las respuestas y construir diálogos inteligentes.

Este lenguaje de pseudo-programación (pues puede prescindir de lenguaje de programación) lo bautizamos **AndyScript**.

Características Internas

El lenguaje **AndyScript**, si bien suena a "script" simplemente se usa un nombre simpático para que se entienda que es un lenguaje fácil de escribir, pero internamente es muy estricto, no admite errores de sintaxis, la mayoría de los cuales son detectados en la fase de análisis 'parsing' inicial.

El lenguaje **AndyScript** es procesado (*parseado y compilado*) activamente como todo lenguaje de computación y combinado luego con los demás paquetes especificados, validado y finalmente el sistema escribe código fuente en C#. Este código es luego linkeado con las librerías internas y externas y compilado por el sistema, (ensamblado a lenguaje de máquina virtual (*MSIL*) posteriormente) para su posterior ejecución, junto a las librerías necesarias. Esto termina en una DLL nativa y usable en .NET en forma directa, conllevando la velocidad de un código de máquina, compilado posteriormente en forma nativa por el JIT del runtime de .NET o Mono, tanto al inicializarse como durante su ejecución.

Estructura General

El sistema armado, que llamaremos agente o bot, consiste en paquetes de código **AndyScript** (*especificaciones de diálogo*) los cuales tienen terminación .PKG; dentro de las cuales se desarrolla toda la sintaxis del sistema. Un sistema o agente funcional consistirá de un conjunto