

Desarrollo de una aplicación para el diálogo en lenguaje natural con un robot móvil

Grigori Sidorov

Laboratorio de lenguaje natural y procesamiento de texto,
Centro de Investigación en Computación (CIC),
Instituto Politécnico Nacional (IPN), México DF,
México

www.cic.ipn.mx/~sidorov

Resumen. En este artículo presentamos tareas de desarrollo y avances de un proyecto que desarrolla todas las fases de la comunicación con un robot móvil en el mundo real. Se integra el software de reconocimiento y síntesis de voz (Dragon Naturally Speaking SDK), el software de manejo de un robot móvil en la arquitectura cliente-servidor, el software de análisis lingüístico (FreeLing), y el software que implementa un modelo de estrategias de diálogo. Las estrategias de diálogo consisten en varios patrones que se obtienen después de la aplicación del análisis lingüístico de las frases de entrada basándose en la detección de las palabras específicas. El software es aplicable al robot móvil tipo *Pioneer* que es un estándar de facto en el campo de la robótica móvil (de los robots que no son de juguete), aunque sin mayores cambios el software puede ser aplicado a otros tipos de robots.

Palabras clave. Robótica móvil, diálogo en lenguaje natural, estrategias de diálogo.

1. Introducción

La robótica móvil es una línea de investigación que tiene un gran potencial en su desarrollo y aplicaciones. En su etapa moderna, cuando ya existen los robots de una buena calidad que permiten su navegación autónoma, es importante poder transmitir los comandos a esos robots a través de la voz, igual que recibir la respuesta del robot a través del mismo medio—la voz, y en general establecer la comunicación natural con el robot..

El uso de la voz permite entablar un diálogo con el robot que imita un diálogo con otro ser humano, y de esa manera la comunicación se vuelve más natural. Para que

este diálogo fuera coherente, aunque se realizase sobre un tema específico—en nuestro caso, sobre los movimientos de mismo robot—, se necesita implementar las siguientes etapas de procesamiento por parte del robot:

- Reconocer la frase que se le manda (la voz),
- Procesar la respuesta en un esquema tradicional de un procesador lingüístico [1],
- Preparar la respuesta según las estrategias de diálogo específicas,
- Generar la frase de respuesta según las reglas gramaticales,
- Pronunciar la respuesta, y
- Realizar la acción solicitada.

En este artículo presentamos un proyecto que desarrolla todas las fases de esta comunicación, integrando:

- Módulo de reconocimiento y síntesis de voz (*Dragon Naturally Speaking SDK*),
- Módulo de manejo de un robot móvil en la arquitectura cliente-servidor,
- Módulo de análisis lingüístico (basado en Freeling¹), y
- Módulo que implemente un modelo de estrategias de diálogo.

Las estrategias de diálogo consisten en varios patrones que se obtienen después de la aplicación del análisis lingüístico de las frases de entrada basándose en la detección de las palabras específicas y sus combinaciones. Eso claramente es una simplificación de un proceso real de diálogo, pero su uso es justificado para los fines prácticos, es decir, para tener un robot reaccionando correctamente a los comandos. Con esos patrones se determina a qué actos de habla corresponden las oraciones y se actúa en consecuencia.

El software es aplicable al robot móvil tipo *Pioneer* que es un estándar de facto en el campo de la robótica móvil, en este caso sin contar los robots del tamaño real, y no los que son de juguete. Sin embargo, sin mayores cambios el software desarrollado puede ser aplicado a otros tipos de robots. En este caso se cambiaría el módulo de la interfaz con el robot de nivel bajo, nada más, es decir, la manera de como transmitir al robot los comandos de movimientos, etc.

El robot móvil tipo *Pioneer* es un robot que se mueve sobre las ruedas, sin embargo, las mismas ideas son aplicables a los robots humanoides (antropomorfos), que utilizan piernas para su movimiento—ya que el hecho de usar piernas en lugar de ruedas claramente no afecta las estrategias de desplazamiento como tal a nivel de planeación y ejecución global.

¹ Freeling es un analizador que resuelve con muy buena calidad muchos problemas de procesamiento automático de lenguaje natural (también para el español), como análisis morfológico, sintáctico, resolución de anáfora, detección de entidades nombradas [19].

2. Descripción del robot

Un robot móvil es un dispositivo que está provisto de los sensores de medición de distancia hasta los obstáculos, tiene las ruedas y la manera muy precisa de medir la distancia recorrida, puede ejercer los comandos de una computadora usando la interfaz serial RS-232, dar vueltas (a un ángulo específico) y moverse hacia atrás o adelante con una velocidad determinada, y obviamente pararse. En sí, no parece que son muchas opciones pero son las características suficientes para modelar el comportamiento relacionado con el movimiento humano en un espacio restringido. En este caso no se usan los módulos de visión, siendo sustituidos por la medición de distancias hasta los obstáculos.

El “cerebro” del robot es una computadora estándar, que realiza todo el procesamiento, y manda al robot los comandos correspondientes. Como es algo que debe moverse junto con el robot y no depender de los cables eléctricos, usualmente esta computadora es un laptop.

Cabe mencionar que el robot tiene dos tipos de sensores: ocho sonares y un sensor laser. El sensor laser obtiene un gran número de lecturas y realmente permite la detección de obstáculos muy precisa, haciendo el uso de los sonares redundante para las aplicaciones prácticas.

Como es usual en el campo de robótica, un robot móvil puede moverse de manera adecuada si tiene de antemano un mapa de su ambiente, entonces una de las tareas previas a su uso es desarrollar este mapa y presentarlo al usuario. Es lógico porque un robot móvil solamente puede medir las distancias hacia los obstáculos; la visión para la robótica móvil es una rama que se está desarrollando rápidamente, pero todavía no tiene los avances necesarios para realizar los movimientos confiables solamente basándose en visión.

Un problema del robot es la imposibilidad de detectar a los obstáculos desnivelados, especialmente los precipicios o escaleras hacia abajo, dado que sus sensores solo “barren” en un plano. Por lo mismo, en el mapa se puede marcar ciertas áreas desniveladas como las áreas prohibidas, y el robot no va a poder pasar por allá y no caería en un precipicio.

En nuestro caso usamos a un robot móvil tipo *Pioneer 3-DX*, aunque el desarrollo es aplicable a todos los robots similares.

3. Los actos de habla

Un diálogo es un intercambio de réplicas de por lo menos dos sujetos. Cada sujeto tiene su modelo del mundo y da un propósito a cada réplica suya, expresando alguna de sus intenciones.

Existen muchos trabajos sobre diálogos en los sistemas automáticos de respuestas, por ejemplo, para consultas de horarios y precios de los boletos, como un ejemplo, mencionamos a Torres (2006) [4]. Sin embargo, nuestro interés principal es un diálogo más complejo, que involucre a un robot.

Roy y Reiter (2005) [9] mencionan algunos problemas fundamentales en la interacción/diálogo con los robots:

- uso de sistemas de representación multimodal,
- asociación de palabras con las acciones y con las percepciones,
- modelado del contexto,
- detección de la granularidad correcta de los modelos,
- integración de modelos y planeación temporales,
- habilidad de mapear las experiencias pasadas, es decir aprendidas, con la interacción actual,
- habilidad de entender y tomar en cuenta la perspectiva de humanos.

En general, podemos afirmar que existen dos tipos de contextos (uno de ellos con dos dimensiones) del diálogo [11]:

- contexto lingüístico,
- contexto extralingüístico:
 - dimensión temporal,
 - dimensión de conocimiento del mundo.

Del otro lado, la estrategia de diálogo puede formalizarse [6, 7] como un proceso de decisión secuencial en términos de mapeo entre posibles estados y posibles acciones, es decir, como una serie de transiciones entre estados a través de acciones. En este caso, la idea sería minimizar el costo de las transiciones entre esos dos conjuntos, donde los nodos son estados del robot y las acciones son las transiciones: para llegar a un estado específico se puede usar varias rutas. Sin embargo, es un problema exponencial. Como siempre en las tareas de optimización exponenciales, se puede buscar la optimización de esas rutas con algún algoritmo de inteligencia artificial, como, por ejemplo, un algoritmo evolutivo.

Una de las posibles soluciones a las estrategias del diálogo mencionadas en [4], también es actual para los robots. Por estrategia de diálogo se entiende el conjunto de decisiones tomadas por el módulo gestor de diálogo que permiten:

- Evitar que el usuario se sienta perdido, explicando las acciones del sistema.
- Responder directamente a las preguntas del usuario.
- Permitir al usuario que corrija al sistema (posibilidad de negociación), por las causas de tanto errores del sistema como por el cambio de decisión del mismo usuario.

En este caso existe la negociación, también llamada la iniciativa mixta, cuando tanto el robot como el humano pueden iniciar la interacción. Cabe mencionar que este trabajo se orienta en sistemas tipo telefónicas, cuando no existe un mundo en común para los sujetos de interacción.

Analizando el lado relacionado con los actos del habla presentado por el creador de esta teoría J. Searle [12]:

- Afirmaciones (*assertives*), actos de habla que afirman la verdad, por ejemplo, profesando.
- Directivas (*directives*), que deben causar alguna acción del oyente: peticiones, comandos, consejos.
- Promesas (*commissives*), que el hablante contrae los compromisos futuros.
- Expresiones (*expressives*) que expresan la emoción o la posición, como felicitaciones, excusas, agradecimientos.
- Declaraciones (*declarations*), que cambian algo en el mundo haciendo una declaración, es decir, que son los performativos.

Se puede observar, que son los actos de habla demasiado generales para su aplicación directa en un modelo práctico de diálogo. Por lo que normalmente se proponen unas clasificaciones más detalladas y específicas.

Otras estrategias generales interesantes más prácticas que pueden ser aplicadas en un sistema específico [14] son:

- Considera las fuentes disponibles de información lo más temprano posible,
- Mantén las hipótesis alternativas y posterga la decisión con una sola opción lo más largo posible.

En la comunicación existen otros principios muy relevantes con respecto a las estrategias conocidas como Máximas de Grice [13], que básicamente son las presuposiciones en una interacción cooperativa exitosa:

1. Máxima de calidad: sé honesto.
 - No digas algo que crees que es falso.
 - No digas algo de que no tienes evidencias.
2. Máxima de cantidad: cantidad de información debe ser adecuada.
 - Haz tú contribución tan informativa como es necesario (para los propósitos dados de intercambio de información).
 - No seas más informativo de lo que es necesario.
3. Máxima de relación (relevancia): sé relevante (es claro intuitivamente para una persona pero es difícil de determinar cómo cumplirlo).
4. Máxima de forma de expresión (manera): sé claro.
 - Evita la obscuridad de expresión.

- Evita la ambigüedad.
- Sé breve.
- Sé ordenado.

Las máximas son aplicables al diálogo, pero son más bien unas reglas generales que modelos prácticos como, por ejemplo, [15, 17]. En un diálogo práctico existen las obligaciones y conocimiento común (*common ground*). Por ejemplo, si el usuario pide algo al sistema, el sistema obtiene la obligación de cumplir eso durante el desarrollo de la interacción, o bien informar que no se puede. Las obligaciones se definen como relaciones entre los diferentes actos de habla.

El conocimiento común garantiza que los participantes comparten el mismo conjunto de creencias e intenciones en cada momento. Esto garantiza que se entenderán correctamente, es decir, en correspondencia a la intención de otra parte. En la práctica a menudo se pierde el conocimiento común y debe restablecerse.

Es necesario recordar manejar correctamente de cambio de turnos, y el cambio o continuación del tópico de conversación. Cabe mencionar que antes de continuar con el otro tema, debe existir la evidencia que de alguna manera terminamos el tema anterior, es conocido como principio de clausura (*principle of closure*).

De manera muy general cualquier diálogo completo sigue al siguiente esquema:

Saludo—Apertura—Negociación/Acción—Finalización—Despedida.

Los actos de habla a menudo pueden tener unos actos de habla de respuesta que les corresponden. Por ejemplo, Levinson [16] menciona las siguientes pares de actos y las reacciones típicas:

- petición → aceptar / rechazar
- oferta → aceptar / rechazar
- invitación → aceptar / rechazar
- evaluación → acuerdo / desacuerdo
- pregunta → respuesta esperada / respuesta inesperada
- acusación → negación / admisión
- disculpa → minimización
- agradecimiento → minimización (ej., “de nada”)
- saludo → saludo

Existen varios modelos de actos de habla en diálogos. Por ejemplo, un modelo muy conocido es DAMSL de Allen y Gore [17], que contiene 4 dimensiones, cada uno incluye algunas subdimensiones y los actos de habla.

1. Estado comunicativo (no operable, abandonado, o monólogo vs. diálogo).
2. Nivel de información.
Actos de habla:

- tarea,
 - mantenimiento de tarea,
 - mantenimiento de comunicación.
3. Análisis hacia adelante (*Forward Looking Function*): afecta las acciones subsecuentes.
- Actos de habla:
- Declaración (*statement*),
 - Compromiso de la acción subsecuente (*committing-speaker-future-action*),
 - Convencional (*conventional*),
 - Performativo (*explicitperformative*),
 - Exclamación (*exclamation*),
 - Otro (lo que dice el hablante): suposición, confirmación,
 - Influencia de la acción subsecuente (*influencing-addressee-future-action*) (como en la teoría de Searle):
 - Una sugerencia o una lista de opciones,
 - Un comando real,
 - Petición de información.
4. Análisis hacia atrás (*Backward Looking Function*): relación con el discurso anterior.
- Actos de habla:
- Concordancia (*agreement*): aceptar, rechazar, mantener comunicación,
 - Entendimiento (*understanding*),
 - Respuesta (*answer*) que corresponde a la petición de información.

Si se trata de los actos de habla de manera más específica, se puede mencionar varios conjuntos de los siguientes actos de habla que se presentan, por ejemplo, en [18]:

- Agradecer (*Thank*)
- Saludar (*Greet*)
- Introducir (*Introduce*)
- Despedirse (*Bye*)
- Solicitar_comentario (*Request_comment*)
- Sugerir (*Suggest*)
- Rechazar (*Reject*)
- Aceptar (*Accept*)
- Solicitar_sugerencia (*Request_suggest*)
- Iniciar (*Init*)
- Argumentar (*Give_reason*)
- Retroalimentar (*Feed_back*)

- Deliberar (*Deliberate*)
- Confirmar (*Confirm*)
- Clarificar (*Clarify*)
- Desviarse (*Digress*)
- Motivar (*Motivate*)
- Desechar (*Garbage*)

Si vamos a considerar alguna clasificación de los actos de habla, una manera de clasificarlos es:

- Toma de turno (*Turn-taking*)
 - Tomar turno (*take turn*)
 - Mantener turno (*keep turn*)
 - Liberar turno (*release turn*)
 - Asignar turno (*assign turn*)
- Mantenimiento (*Grounding*)
 - Confirmar (*acknowledge*)
 - Reparar (*repair*)
 - Continuar (*continue*)
- Básico (*Core*)
 - Informar (*inform*)
 - Preguntar algo específico (*wh-question*)
 - Aceptar (*accept*)
 - Requerir (*request*)
 - Ofrecer (*offer*)
- Argumentación (*Argumentation*)
 - Elaborar (*elaborate*)
 - Resumir (*summarize*)
 - Pregunta-respuesta (*question-answering*)
 - Clarificar (*clarify*)

4. Tareas de desarrollo

Como ya mencionamos anteriormente, el objetivo general de nuestro desarrollo se puede formular de la siguiente manera: desarrollar un software que permita dialogar con un robot móvil sobre sus movimientos utilizando el lenguaje natural. La implementación del software incluye la integración de todos los módulos, lo que según nuestra experiencia no es una tarea trivial. Nótese que es muy difícil para un robot de “entender” mensajes (textos) en profundidad, como por ejemplo, realizar el procesamiento semántico tipo, digamos, *textual entailment* [5].

Cabe mencionar que el software desarrollado funciona utilizando tres capas respecto a la comunicación con el robot. De hecho es un diseño relativamente estándar. Existe una capa de bajo nivel que se encarga de la interacción física con el robot móvil. La programación de esta capa obviamente depende del tipo del robot que estamos utilizando. Otra capa es un programa servidor, que cumple con las funciones de enlace entre el cliente y la capa de bajo nivel. En este sentido, el programa servidor recibe las instrucciones de cliente en un nivel bastante general, por ejemplo, “dar vuelta a la derecha”, o “pararse”, y los codifica (transforma) en los comandos de nivel bajo del robot. El programa cliente se encarga del diálogo con el usuario, presentándole la posición actual del robot en el mapa, mostrándole el progreso de movimientos del robot, presentando las intenciones de los movimientos del robot, procesando los comandos del usuario (realizando tanto el reconocimiento de voz como el análisis lingüístico), generando las respuestas y contestando al usuario (síntesis de voz).

Desde el punto de vista técnico más detallado, la implementación incluye varios submódulos.

1. Desarrollo del parte de servidor que corre en el robot² y recibe los comandos del cliente y los transmite al robot móvil físicamente.
2. Desarrollo del parte de cliente que genere los comandos para el robot móvil y los transmite al servidor.
3. Desarrollo del parte de cliente: desarrollo de la interfaz del usuario para visualizar la interacción con el robot.
4. Desarrollo del parte de cliente: integración del módulo de reconocimiento de voz (librerías de *Dragon SDK*).
5. Desarrollo del parte de servidor y/o cliente: integración del módulo de síntesis de voz (librerías de *Dragon SDK*).
6. Desarrollo del parte de cliente: desarrollo del módulo de análisis lingüístico de las frases de entrada (análisis morfológico y/o sintáctico).
7. Desarrollo del parte de servidor y/o cliente: desarrollo del módulo de generación de las respuestas que corresponden a la situación.
8. Desarrollo del parte de cliente: desarrollo del módulo de estrategias de diálogo, es decir, basándose en la frase de entrada, qué tipo de frase de salida se debe producir y qué acción se debe tomar.

Dentro de cada módulo debemos mencionar las siguientes funciones de los submódulos que se desarrollaron:

1. Desarrollo del parte de servidor que corre en el robot y recibe los comandos del cliente y los transmite al robot móvil físicamente.

² Como ya mencionamos, el robot en este contexto es la computadora que controla al robot.

- Investigación de las opciones posibles disponibles en librerías de *MobileRobots* SDK para la interacción con el robot.
 - Desarrollo de la interfaz del servidor.
 - Desarrollo de la parte que corresponde a la interacción usando el protocolo TCP IP.
2. Desarrollo del parte de cliente que genere los comandos para el robot móvil y los transmite al servidor.
 - Desarrollo de la parte que corresponde a la interacción usando el protocolo TCP IP.
 - Determinación de los comandos que pueden ser interpretados por el servidor.
 3. Desarrollo del parte de cliente: desarrollo de la interfaz del usuario para visualizar la interacción con el robot.
 - Desarrollo de los requisitos de la información requerida por el usuario,
 - Determinación de los parámetros importantes para el comportamiento del robot (robot tiene más de 100 parámetros que cambian su comportamiento, por ejemplo, que tanta gente puede estar en su espacio para que no se considere perdido, velocidad máxima permitida, etc.).
 - Elaboración de la lista de valores recomendados de los parámetros, con opciones de varios entornos.
 - Desarrollo de la representación de los parámetros en la interfaz.
 - Desarrollo de la representación del robot y de su ambiente (el mapa).
 - Desarrollo del manual de elaboración del mapa de ambiente (que es una parte indispensable para el funcionamiento del robot).
 4. Desarrollo del parte de cliente: integración del módulo de reconocimiento de voz (librerías de *Dragon* SDK).
 - Desarrollo de una aplicación de prueba.
 - Desarrollo de un manual de entrenamiento del sistema de reconocimiento (tiene duración alrededor de 5 minutos).
 - Investigación del funcionamiento del sistema sin entrenamiento.
 - Integración de los parámetros del sistema en la interfaz del usuario.
 5. Desarrollo del parte de servidor y/o cliente: integración del módulo de síntesis de voz (librerías de *Dragon* SDK).
 - Desarrollo de una aplicación de prueba.
 - Integración del módulo en la interfaz del usuario.
 - Investigación de la posibilidad de utilizar el apoyo visual al momento del habla (una cabeza parlante en la pantalla).
 6. Desarrollo del parte de cliente: desarrollo del módulo de análisis lingüístico de las frases de entrada (análisis morfológico y/o sintáctico).

- Desarrollo del módulo de análisis morfológico de frase de entrada.
 - Incorporación del análisis sintáctico a través de módulo de análisis sintáctico de *Freeling* [19].
7. Desarrollo del parte de servidor y/o cliente: desarrollo del módulo de generación de las respuestas que corresponden a la situación.
 - Desarrollo de patrones de generación de las respuestas.
 - Desarrollo del módulo de la generación morfológica para los patrones.
 8. Desarrollo del parte de cliente: desarrollo del módulo de estrategias de diálogo, es decir, basándose en la frase de entrada, qué tipo de frase de salida se debe producir y qué acción se debe tomar (gestor de diálogo).
 - Desarrollo de las correspondencias entre las estructuras de frases, su contenido léxico y las situaciones.
 - Detección de las palabras que son marcadores de algunas situaciones.
 - Desarrollo de las estrategias de diálogo, es decir, basándose en la frase de entrada, qué tipo de frase de salida se debe producir y qué acción se debe tomar.

5. Ejemplo de patrones

Vamos a ver un ejemplo. Como una aproximación podemos fijar los siguientes actos de comunicación para los casos más simples:

- Informar_interrupción,
- Informar_estado,
- Informar_falla_de_proceso,
- Informar_exito,
- Informar_falla_de_comunicación

Los actos de comunicación corresponden a las respuestas que se generan por el robot y se determinan por su estado interno. Se puede considerarlos como unas funciones dentro de las cuales se realiza la sub-respuesta (una respuesta que forma parte de la otra más compleja).

Después de realizar el análisis sintáctico, se determina el verbo principal de la oración. Se determinan los verbos que corresponde a los movimientos: *moverse*, *cambiarse*, *ir*, *llegar*, *avanzar*, *desplazarse*, etc. De esa manera:

VERBO = el verbo principal de la oración,

NP_{meta} = una frase nominal que se encuentra en el mapa y corresponde a una meta.

Podemos utilizar el siguiente metalenguaje para describir la comunicación.

IF (VERBO en ‘verbos de movimiento’ Y NP en METAS)

```
// (VERBO está en la lista de ‘verbos de movimiento’ Y NP en las METAS del
mapa actual)
THEN
1) INFORMAR_INTERRUPCIÓN (META_ACTUAL)
2) EXITO = PLANEAR_RUTA (META_NUEVA)
3) IF EXITO
    THEN
        4) INFORMAR_EXITO (planear, META_NUEVA)
        5) INFORMAR_ESTADO (META_NUEVA)
    ELSEIF
        6) INFORMAR_FALLA ()
ENDIF
ENDIF
```

Por el momento no proporcionamos la descripción detallada del lenguaje, pero creemos que para cualquier persona que tiene conocimientos básicos de programación es bastante claro lo que sucede dentro de este pseudocódigo.

6. Conclusiones

En este artículo hemos presentado las tareas de desarrollo para el diálogo en lenguaje natural con un robot móvil. Para que este diálogo con el robot sea coherente, aunque fuera sobre un tema específico (en este caso, sobre los movimientos del robot), se necesita que el robot reconozca la frase que se le manda (la voz), la procese, prepare la respuesta según las estrategias de diálogo específicas, genere la frase de respuesta, la pronuncie, y realice la acción solicitada.

Este proyecto desarrolla todas las fases de esta comunicación, integrando el software de reconocimiento y síntesis de voz (*Dragon Naturally Speaking* SDK), el software de manejo de un robot móvil en la arquitectura cliente - servidor, y el software de análisis lingüístico, mismo que el software que implemente un modelo de estrategias de diálogo. Las estrategias de diálogo consisten en varios patrones que se obtienen después de la aplicación del análisis lingüístico de las frases de entrada basándose en la detección de las palabras específicas. El software es aplicable al robot móvil tipo *Pioneer* (que es un estándar de facto en el campo de la robótica móvil de los robots que no son de juguete), aunque sin mayores cambios el software puede ser aplicado a otros tipos de robots.

Agradecimientos. Trabajo realizado con el apoyo de gobierno de la Ciudad de México (proyecto ICYT-DF PICCO10-120), el apoyo parcial del gobierno de México (proyectos CONACYT 50206-H y 83270, SNI) e Instituto Politécnico Nacional, México (proyectos SIP 20111146, 20113295, 20120418, COFAA).

Referencias

1. Gelbukh, A., Sidorov, G.: Procesamiento automático del español con enfoque en recursos léxicos grandes. IPN, 307 p. (2010)
2. Dialog with Robots. AAAI 2010. Fall Symposium. November 2010, Arlington VA. Available: http://hci.cs.wisc.edu/aaai10/?page_id=169 (2010)
3. Lemaignan, S. and Ros, R. and Alami R.: Dialogue in situated environments: A symbolic approach to perspective-aware grounding, clarification and reasoning for robot. In: Proc. Robotics, Science and Systems, Grounding Human-Robot Dialog for Spatial Tasks workshop (2011)
4. Torres Goterris, F.: Sistemas de diálogo basados en modelos estocásticos. PhD Thesis, UPV, Valencia (2006)
5. Pakray, P., Barman, U., Bandyopadhyay, S., Gelbukh, A.: A Statistics-Based Semantic Textual Entailment System.. MICA 2011, Lecture Notes in Artificial Intelligence N 7094, Springer, pp. 267–276 (2011)
6. Levin, E., Pieraccini, R., Eckert, W., Di Fabbrizio, G., Narayanan, S.: Spoken language dialogue: from theory to practice. In: Proc. of ASRU–IEEE Workshop. Keystone, Colorado, USA (1999)
7. Levin, E., Pieraccini, R., Eckert, W.: A stochastic model of human-machine interaction for learning dialog strategies. IEEE Transactions on Speech and Audio Processing. vol. 8 (1), pp. 11–23 (2000)
8. Marge, M., Pappu, A., Frisch, B., Harris, Th. K., Rudnick, A.: Exploring Spoken Dialog Interaction in Human-Robot Teams. In: Proceedings of Robots, Games, and Research: Success stories in USARSim IROS Workshop, October 2009, St. Louis, MO, USA (2009)
9. Roy, D., Reiter, E.: Connecting language to the world. Artificial Intelligence (2005)
10. Sisbot, E.A., Ros, R., Alami, R.: Situation assessment for human-robot interaction. In: Proc. of 20th IEEE International Symposium in Robot and Human Interactive Communication (2011)
11. Kruijff, G.J.M., Lison, P., Benjamin, T., Jacobsson, H., Zender, H., Kruijff-Korabayova, I., Hawes, N. Situated dialogue processing for human-robot interaction. Cognitive Systems, pp. 311–364 (2010)
12. Searle, J.: Indirect speech acts. In: Syntax and Semantics, 3: Speech Acts, ed. P. Cole & J. L. Morgan, pp. 59–82. New York: Academic Press (1975) [Reprinted in Pragmatics: A Reader, ed. S. Davis, pp. 265–277. Oxford: Oxford University Press (1991)]
13. Grice, P.: Logic and conversation. In: Syntax and Semantics, 3: Speech Acts, ed. P. Cole & J. Morgan, New York: Academic Press (1975) [Reprinted in Studies in the Way of Words, ed. H. P. Grice, pp. 22–40. Cambridge, MA: Harvard University Press (1989)]
14. Souvignier, B., Kellner, A., Rueber, B., Schramm, H., Seide, F.: The Thoughtful Elephant: Strategies for Spoken Dialog Systems. IEEE Transactions on Speech and Audio Processing. vol. 8 (1), pp. 51–62 (2000)

15. Pineda, L.A., Estrada, V.M., Coria, S.R., Allen, J.F.: The obligations and common ground structure of practical dialogues. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 11(36), pp. 9-17 (2007)
16. Levinson, S.C. *Pragmatics*, Cambridge University Press, Cambridge, MA, USA (1983)
17. Allen, J.F., Core, M.G.: DAMSL: Dialog Act Markup in Several Layers (Draft 2.1), Multiparty Discourse Group, Discourse Research Initiative (1997)
18. Jurafsky, D., Martin, J.: *Speech and Language Processing*. Prentice Hall (2009)
19. Padró, L., Collado, M., Reese, S., Lloberes, M., Castellón, I.: FreeLing 2.1: Five Years of Open-Source Language Processing Tools. In: *Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010)*, ELRA La Valletta, Malta (2010)