

Curso: Generación de Lenguaje Natural y Aplicaciones

Escuela de Lingüística Computacional 2010

26-31 de Julio 2010, Buenos Aires, Argentina

Evaluación: Proyectos

Entrega: **Lunes 23 de Agosto, 2010**

1 Instrucciones Generales para Todos los Proyectos:

El proyecto consiste en extender un aspecto (el aspecto que hayan elegido como indica la Sección 2) de un generador de instrucciones GIVE. Los pasos a seguir son los siguientes:

1. Instalar el framework: Las instrucciones para instalar el framework GIVE están en <http://kenai.com/projects/give-challenge/pages/Manual#Configuration>. Para asegurar la compatibilidad de las partes del framework instalen la versión más reciente de cada una de las partes.
2. Modificar el código del Servidor de NLG: Para este proyecto tienen que tomar el código del `give2-example-nlgserver` y modificarlo con el aspecto solicitado en la Sección 2, no tienen que comenzar desde cero con un nuevo sistema NLG. La documentación en http://kenai.com/projects/give-challenge/pages/Manual#Implementing_your_own_NLG_server (que explica cómo implementar un Servidor NLG desde cero) puede ser útil para entender cómo funciona el `give2-example-nlgserver`. Quizás también quieran consultar el API-doc que está en <http://www.give-challenge.org/give2/apidocs/>.
3. Testear en los siguientes 3 mundos diferentes:
 - Un mundo simple:
<http://www.glyc.dc.uba.ar/elic2010/development-world.giveworld>
 - Un mundo con varios distractores:
<http://www.glyc.dc.uba.ar/elic2010/evalworld-1-2010.giveworld>
 - Un mundo con navegación compleja:
<http://www.glyc.dc.uba.ar/elic2010/testworld3.giveworld>

Para cambiar de mundo es necesario modificar el archivo de configuración del matchmaker, es decir `matchmaker-config.xml` como se explica en:

http://kenai.com/projects/give-challenge/pages/Manual#Configuring_the_Matchmaker

4. Escribir un informe de 8 páginas (al menos) describiendo lo que hicieron y cómo. Incluyendo ejemplos de situaciones interesantes en los mundos (recomendamos el uso de screenshots para ilustrar las situaciones).

El proyecto se entrega por mail el **Lunes 23 de Agosto** a carlos.areces@gmail.com. Envíen **pdf** del informe (de latex, doc, scan, etc), más el **código modificado** e instrucciones de cualquier cosa que debamos modificar en el framework (aunque esto no debería ser necesario para lo que se pide en el proyecto). Para asegurarnos que el proyecto está en marcha les ofrecemos 2 ‘check points’; preguntas sobre esos temas (por email a carlos.areces@gmail.com) serán contestados en el fin de semana correspondiente.

- Viernes 6: Problemas de Instalación
- Viernes 13: Problemas de Implementación

2 Descripciones y asignaciones de proyectos

A continuación se describen los proyectos en detalle. El objetivo de estas descripciones es darles ejemplos concretos de lo que se requiere para aprobar el proyecto. Si durante el desarrollo del proyecto deciden incluir otras extensiones, adelante!. Si quieren modificar alguna de las extensiones que nosotros les proponemos (por el motivo que sea) ponganse en contacto con nosotros primero.

Proyecto 1: Determinación de contenido - Cambio de granularidad

Asignado a: Cossio Christian y Martínez Miguel

Descripción: El objetivo de este proyecto es modificar la forma en la que realiza la determinación del contenido a transmitir el sistema `give2-example-nlgserver`. Este sistema toma el plan y verbaliza cada una de las acciones del plan como una instrucción, es decir, usa un nivel de granularidad 1 para la determinación del contenido (usando la terminología que introducimos en clase). Por ejemplo, al comenzar la interacción en el mundo `development-world` dice lo siguiente:

```

NLG: Look around until you see a blue thing. Then go towards it.
User: [goes really close to the button]
NLG: Excellent! Press the button.
User: [walks backwards a bit to center the button]
NLG: Wait. That's not what I wanted you to do. [replan] Go to the
button.
User: [presses the button]
NLG: Wait. That's not what I wanted you to do. [replan] Go to the door.
User: [goes through the door]
NLG: Excellent!

```

Modifique el sistema `give2-example-nlgserver` para que use un nivel de granularidad 3 para la determinación del contenido y para monitorear la ejecución. En el ejemplo anterior debería generar lo siguiente:

```

NLG: Press the blue button.
User: [goes really close to the button]
User: [walks backwards a bit to center the button]
User: [presses the button]
NLG: Great!
NLG: Look for the blue room.
User: [goes through the door]
NLG: Great!

```

Es decir, el nuevo Servidor NLG no debe replanear si el usuario se sale del plan en el nivel 1. El sistema debe replanear sólo cuando el usuario presiona el botón incorrecto.

Proyecto 2: Determinación de contenido - Landmarks en navegación

Asignado a: Daniel Koile y Celeste Gunski

Descripción: El objetivo de este proyecto es modificar la forma en la que realiza la determinación del contenido a transmitir el sistema `give2-example-nlgserver`. El sistema actualmente no usa objetos con los que el usuario no puede interactuar (llamados landmarks) para guiar la navegación. Este proyecto consiste en agregar esa capacidad. Por ejemplo, al comenzar la interacción en el mundo `evalworld-1-2010.giveworld` dice lo siguiente:

NLG: Go straight.
User: [goes straight towards a chair]
NLG: Excellent! Look around until you see a button near a green thing,
NLG: then go towards it.
NLG: It should be in front of you slightly to the right.
User: [turns right]
User:[goes straight towards the red button by a plant]
NLG: Excellent! Press the button

Modifique el sistema `give2-example-nlgserver` para que use landmarks (plantas, lamparas, sillones, etc) durante la determinación del contenido de instrucciones de movimiento y para monitorear la ejecución. Es decir, el sistema debe analizar las instrucciones de movimiento `move(from, to)` y comprobar si la región `to` contiene un landmark. En el ejemplo anterior debería generar lo siguiente:

NLG: Go towards the plant.
User: [goes towards the plant]
NLG: Excellent! Look around until you see a button near a green thing,
NLG: then go towards it.
NLG: It should be in front of you slightly to the right.
User: [turns right]
User:[goes straight towards the red button]
NLG: Excellent! Press the button

Puede ocurrir, como es el caso del ejemplo anterior, que hay más de un landmark en la región `target`; en el ejemplo hay una lámpara, una silla y una planta. En este caso, el sistema debe elegir el landmark que está **más cerca** de la región `to` de la siguiente acción `move`; en el ejemplo la planta es la que está más cerca de la región del botón que hay que presionar. Nótese que el monitoreo de las instrucciones que incluyen landmarks también debe modificarse para que el éxito de la acción no ocurra cuando el usuario entre en la región que tiene el landmark sino cuando el usuario esté cerca del landmark (deberá definir qué considera *cerca* en este contexto).

Proyecto 3: Generación de expresiones referenciales con deducción por eliminación **Asignado a: Etcheverry Mathias y Zeballos Yasim**

Descripción: El objetivo de este proyecto es modificar la forma en la que realiza el proceso de monitoreo de expresiones referenciales generadas el sistema `give2-example-nlgserver`. El sistema actualmente no monitorea la identificación de los objetos `target` de una expresión referencial generada. Este proyecto consiste en agregar esa capacidad. Por ejemplo, al comenzar la interacción en el mundo `evalworld-1-2010.giveworld` dice lo siguiente:

NLG: Go straight.
User: [goes straight towards a chair]
NLG: Excellent! Look around until you see a button near a green thing,
NLG: then go towards it.
NLG: It should be in front of you slightly to the right.
User: [turns right]
User:[goes straight towards the red button by a plant]
NLG: Excellent! Press the button

Modifique el sistema `give2-example-nlgserver` para que monitoree la identificación de objetos `target` de una expresión referencial. Por ejemplo, el objeto `target` de la expresión referencial `button near a green thing` es el botón rojo al lado de la planta. El sistema modificado debe indicar que ese es el botón `target` cada vez que se hace visible y que otros botones rojos en la misma habitación no lo son. En el ejemplo anterior debería generar lo siguiente:

NLG: Go straight.
User: [goes straight towards a chair]
NLG: Excellent! Look around until you see a button near a green thing,
NLG: then go towards it.
NLG: It should be in front of you slightly to the right.
User: [walks around and sees a red button near a lamp]
NLG: Not that one!
User: [walks around and sees a red button near a picture]
NLG: Not that one!
User: [walks around and sees the target red button next to a plant]
NLG: Yeah! That one!
User:[goes straight towards the red button]
NLG: Excellent! Press the button

Proyecto 4: Generación de expresiones referenciales relacionales (relación con landmark o distractor)

Asignado a: Navarro Andrés y Fontana Fernando

Descripción: El objetivo de este proyecto es modificar la forma en la que realiza el proceso de generación de expresiones referenciales el sistema `give2-example-nlgserver`. El sistema actualmente no genera expresiones referenciales como `the yellow button on the left of the picture`. Este proyecto consiste en agregar esa capacidad. Por ejemplo, al comenzar la interacción en el mundo `evalworld-1-2010.giveworld` dice lo siguiente:

NLG: Go straight.
User: [goes straight towards a chair]
NLG: Excellent! Look around until you see a button near a green thing,
NLG: then go towards it.
NLG: It should be in front of you slightly to the right.
User: [turns right]
User:[goes straight towards the red button by a plant]
NLG: Excellent! Press the button
User:[presses the button]
NLG: Great! Turn left almost all the way but not quite. Then go in that direction.
User: [turns and goes forward]
NLG: Excellent! Go to the button near the red button.
NLG: It should be in front of you.

Modifique el sistema `give2-example-nlgserver` para que prefiera en uso de landmarks (objetos de tipo diferente al tipo del target de la expresión referencial, e.g. cuadros, cómodas, etc), en lugar de distractores (objetos del mismo tipo, como botones), durante la generación de expresiones referenciales. En el ejemplo anterior podría generar lo siguiente:

NLG: Go straight.
User: [goes straight towards a chair]
NLG: Excellent!
NLG: Look around until you see a red button on the left of the closed door,
NLG: then go towards it.
NLG: It should be in front of you slightly to the right.
User: [turns right]
User:[goes straight towards the red button by a plant]
NLG: Excellent! Press the button
User:[presses the button]
NLG: Great! Turn left almost all the way but not quite. Then go in that direction.
User: [turns and goes forward]

NLG: Excellent! Go to the yellow button on the left of the picture.

NLG: It should be in front of you.

La extensión se pide para landmarks que se encuentran apoyados en las paredes (como cuadros, cómodas, puertas y camas) y no necesariamente para otros landmarks que pueden estar a la derecha o a la izquierda de un botón dependiendo de la ubicación del jugador (como plantas).

Proyecto 5: Generación de expresiones referenciales centradas en el usuario

Asignado a: Guerrero Augusto César, Ossola María Florencia y Orellaia Luciana

Descripción: El objetivo de este proyecto es modificar la forma en la que realiza el proceso de generación de expresiones referenciales el sistema `give2-example-nlgserver`. El sistema actualmente no genera expresiones referenciales como `the yellow button furthest from you`. Este proyecto consiste en agregar esa capacidad. Por ejemplo, al comenzar la interacción en el mundo `evalworld-1-2010.giveworld` dice lo siguiente:

NLG: Go straight.

User: [goes straight towards a chair]

NLG: Excellent! Look around until you see a button near a green thing,

NLG: then go towards it.

NLG: It should be in front of you slightly to the right.

User: [goes straight towards the red button by a plant]

NLG: Excellent! Press the button

User: [presses the button]

NLG: Great! Turn left almost all the way but not quite. Then go in that direction.

User: [turns and goes forward]

NLG: Excellent! Go to the button near the red button.

NLG: It should be in front of you.

Modifique el sistema `give2-example-nlgserver` para que use la distancia del target de la expresión referencial con respecto al jugador como una propiedad a incluir en la expresión referencial. En el ejemplo anterior podría generar lo siguiente:

NLG: Go straight.

User: [goes straight towards a chair]

NLG: Excellent! Look around until you see the red button furthest from you,

NLG: then go towards it.

NLG: It should be in front of you slightly to the right.

User: [goes straight towards the red button by a plant]

NLG: Excellent! Press the button

User: [presses the button]

NLG: Great! Turn left almost all the way but not quite. Then go in that direction.

User: [turns and goes forward]

NLG: Excellent! Go to the yellow button furthest from you.

NLG: It should be in front of you.