

SimpleNLG-es

Tabla de contenido

Introducción.....	1
Marco de control.....	2
NLGElement.....	2
Atributos.....	2
Métodos.....	2
NLGModule.....	4
Atributos.....	4
Métodos.....	4
ElementCategory.....	5
Método.....	5
Lexicon.....	5
Clases.....	5
Lexicon.....	5
XMLLexicon.....	5
Módulos.....	6
Módulo sintáctico.....	6
Módulo ortográfico.....	6
Módulo morfológico.....	6
Módulo de agregación.....	6
Módulo Realizador predeterminado.....	6
Módulos de formateo.....	6

Introducción

El propósito de este documento es dar una visión general del concepto de diseño inicial para la arquitectura de SimpleNLG-es. La documentación principal para la implementación consiste en el JavaDoc generado.

Marco de control

El marco de control contiene las clases base y interfaces que definen lo que significa ser un módulo de SimpleNLG o un elemento básico. Tener una definición de elemento básico garantiza que todos los módulos tengan las mismas expectativas sobre cómo son los datos, mientras que tener una definición base de un módulo permite a los módulos ejecutar otros módulos.

Las clases e interfaces definidas se encuentran en el paquete *simplenlg.framework*.

NLGElement

NLGElement es una clase abstracta que representa a todos los componentes del lenguaje. Esto abarca tanto los componentes sintácticos como los componentes estructurales de los documentos, dos ejemplos son la frase nominal (sintaxis) y el párrafo (estructura del documento). Un NLGElement puede referirse a otros elementos en forma de árbol. Así, un elemento de documento de nivel superior puede contener una serie de secciones, que a su vez conducen más abajo en el árbol a párrafos, oraciones, varios tipos de frases y, finalmente, a las propias palabras.

Atributos

TIPO	NOMBRE	DEFINICIÓN
String	baseForm	Para palabras, esto representa la forma base de la palabra utilizada antes de que haya sido modificada por algún módulo.
ElementCategory	category	Define el tipo de NLGElement que es.
List<NLGElement>	complements	Estos son los componentes hijo de este elemento. Los complementos unidos forman este tipo de elementos. Por ejemplo, en una frase los complementos podrían representar las palabras individuales.
HashMap<String, Object>	features	Una lista de las características relevantes para este NLGElement. No puede haber dos características con el mismo nombre sin embargo el valor puede ser una colección o una matriz.
HashSet<String>	flags	Los atributos booleanos podrían agregarse al mapa de características como se mencionó anteriormente. Sin embargo, este campo tiene una forma más conveniente de almacenar los valores booleanos. Sólo los indicadores que son true se almacenan en el conjunto.
NLGElement	head	Usado principalmente en las frases, representa el elemento principal.
NLGElement	parent	Una referencia al nodo padre en el <i>árbol</i> .
String	realisation	La realización actual del elemento. En otras palabras, la representación textual del elemento.

Métodos

SALIDA	NOMBRE	PARÁMETROS	DEFINICIÓN
--------	--------	------------	------------

void	addComplement	NLGElement	Añade un elemento <i>hijo</i> a este elemento.
void	clearAllComplements		Elimina todos los elementos <i>hijo</i> de este elemento.
void	clearAllFeatures		Elimina todas las características de este elemento.
void	clearAllFlags		Elimina todos los flags de este elemento, estableciendo de hecho los flags a false .
String	getBaseForm		Devuelve la forma base de la palabra.
ElementCategory	getCategory		Devuelve el tipo de este elemento.
List<NLGElement>	getComplements		Devuelve los elementos <i>hijo</i> de este elemento.
Object	getFeature	String	Devuelve el valor de la característica como un objeto.
NLGElement	getFeatureAsElement	String	Devuelve el valor de la característica como un NLGElement.
String	getFeatureAsString	String	Devuelve el valor de la característica como una cadena de texto.
NLGElement	getHead		Devuelve el elemento principal de este elemento.
NLGElement	getParent		Devuelve el elemento padre de este elemento.
String	getRealisation		Devuelve la realización actual de este elemento.
boolean	hasFeature	String	Determina si este elemento tiene un valor asociado a una característica.
boolean	isA	ElementCategory	Determina si este elemento es del tipo indicado.
boolean	isFlagTrue	String	Determina si la flag indicada es true .
void	removeComplement	NLGElement	Elimina el <i>hijo</i> indicado de este elemento.
void	removeFeature	String	Elimina la característica indicada de este elemento.
void	removeFlag	String	Elimina la flag indicada de este elemento.
void	setBaseForm	String	Establece la forma base de este elemento.
void	setCategory	ElementCategory	Establece el tipo de este elemento.
void	setComplement	NLGElement	Elimina todos los elementos <i>hijo</i> existentes y añade sólo a este <i>hijo</i> .
void	setFeature	String, Object	Establece la característica indicada de este elemento con el objeto dado.
void	setFlag	String, boolean	Establece la flag indicada de este elemento con el valor booleano dado.
void	setHead	NLGElement	Establece el elemento principal de este elemento.
void	setParent	NLGElement	Establece el elemento padre de este elemento.
void	setRealisation	NLGElement	Establece la realización actual de este elemento.

NLGModule

NLGModule es una clase abstracta que define un módulo individual del framework. Todos los módulos implementados deben extender esta clase. Es posible enlazar los módulos mediante la adición de módulos a los módulos existentes. Los módulos añadidos se ejecutarán antes del propio módulo principal. Los módulos se pueden agregar en función de la prioridad, donde el módulo con el valor más bajo de prioridad se ejecuta antes de cualquier otro módulo.

Atributos

TIPO	NOMBRE	DEFINICION
static final String	SALIENCE_MODIFER	El modificador predeterminado para los valores de prioridad utilizados principalmente al agregar nuevos módulos al principio o al final de la lista de módulos actuales. El valor es 100.
TreeMap<Integer, NLGModule>	modules	La lista de módulos que se deben ejecutar antes de ejecutar este módulo.

Métodos

SALIDA	NOMBRE	PARÁMETROS	DEFINICIÓN
int	addModule	NLGModule	Agrega un módulo al final del árbol de los módulos actuales. Se devuelve el valor de prioridad aplicado.
NLGModule	addModule	NLGModule, int	Inserta un módulo en el árbol en el punto indicado por su prioridad. Si el árbol ya contiene un módulo con esa prioridad se sobrescribe y se devuelve el módulo antiguo.
void	clearModules		Elimina todos los módulos del árbol.
Collection<NLGModule>	getModules		Devuelve la colección de módulos asociados con este módulo en orden de prioridad.
abstract void	initialise		Permite al módulo realizar alguna inicialización antes de ejecutarse.
abstract String	realise	List<NLGElement>	Hilo de procesamiento principal del módulo que realiza las acciones de la lista de NLGElements dada.
abstract String	realise	NLGElement	Hilo de procesamiento principal del módulo que realiza las acciones del NLGElement dado.
NLGModule	removeModule	int	Elimina el módulo con la prioridad indicada del árbol. Se devuelve el módulo eliminado.
int	removeModule	NLGModule	Elimina el módulo indicado del árbol de módulos. Se devuelve la prioridad del módulo eliminado. Si no existiera dicho módulo, se devuelve Integer.MIN_VALUE.

ElementCategory

ElementCategory es una interfaz que debe ser implementada por las clases o tipos enumerados utilizados para definir las categorías sintácticas o estructurales. Tener esta interfaz permite utilizar varios tipos enumerados (como para tener una separación entre la estructura sintáctica y la estructura del documento), asegurando al mismo tiempo que los módulos tengan un acceso genérico a todos estos tipos.

Método

SALIDA	NOMBRE	PARÁMETROS	DEFINICIÓN
boolean	equalTo	Object	Determina si el objeto indicado es igual a esta categoría.

Lexicon

El lexicon es una base de datos de información sobre palabras. Los datos sobre cada palabra incluyen:

- Forma base (p. ej. "perro")
- Categoría sintáctica (p. ej. "sustantivo")
- Características. Esto incluye:
 - Información morfológica (tipo de inflexión, formas irregulares)
 - Información sintáctica (complementos, posición adj, etc.)

Clases

Lexicon

Interfaz/clase genérica que permite a los usuarios:

- Obtener una palabra, dada su forma base (y opcionalmente categoría sintáctica).
- Obtener una palabra dada su ID.
- Obtener una palabra dada una forma flexionada (por ejemplo, "perros" en lugar de "perro") y categoría sintáctica.

XMLLexicon

Clase para cargar palabras a partir de un archivo XML. Todo el archivo se lee en el momento de la carga.

Módulos

Módulo sintáctico

El módulo sintáctico se utiliza para construir la sintaxis del texto mediante la creación de diferentes tipos de frases, p. ej. frase nominal, frase verbal, frase adjetiva. Realizar una llamada de realización sobre estos elementos devolverá la forma base de todos los elementos.

Aparte de ser una extensión de la clase NLGModule, el procesador sintáctico también contiene una serie de clases de soporte para definir el lenguaje, tales como una serie de tipos enumerados para representar características particulares (género, aspecto, tiempo). Todos ellos aparecerán bajo `nlg.syntax.english`. Debería ser posible crear sintaxis adicional para otros idiomas.

Módulo ortográfico

El módulo ortográfico toma el texto básico y añade la puntuación y las mayúsculas. Por ejemplo, "el mensajero entregó la bicicleta verde a María" se convertiría en "El mensajero entregó la bicicleta verde a María". La primera letra de la oración se pone en mayúsculas y la oración se termina con un punto. Estas acciones se realizan al llamar al método de realización.

Módulo morfológico

El módulo Morfológico controla la forma correcta de las palabras. Se encarga de la pluralización y del cambio de tiempo entre las palabras. Por ejemplo, la oración "María persigue a Carlos." está en tiempo presente, si queremos cambiarlo a tiempo pasado, el módulo de morfología convertiría "persigue" a "persiguió" por lo que la frase se convertiría en "María persiguió a Carlos."

Módulo de agregación

El módulo de agregación unirá las oraciones juntas, típicamente mediante el uso de 'y', 'o' y 'pero'.

Módulo Realizador predeterminado

El módulo realizador predeterminado enlaza los módulos antes mencionados. Un usuario puede construir sus elementos y ejecutar el realizador. Este llamará a cada módulo a su vez para producir la cadena final completamente realizada.

Módulos de formateo

Hay dos tipos de módulos de formateo en el paquete SimpleNLG. Uno de los módulos añade las etiquetas HTML al texto devuelto, p. ej. etiquetas de párrafo "<p>" y listas "". El segundo módulo realiza un formateo de texto simple, incluyendo lo siguiente:

- Añadir el título del documento al principio del texto.
- Añadir títulos de sección en los lugares correctos.
- Agregar saltos de línea para facilitar la lectura.
- Agregar elementos de lista con '*'.

- Agregar números para listas enumeradas (por ejemplo, "1.1 -", "1.2 -", etc.).