

SimpleNLG-gl

Táboa de contido

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

Introdución

O propósito deste documento é dar unha visión xeral do concepto de deseño inicial para a arquitectura de SimpleNLG-gl. A documentación principal para a implementación consiste no JavaDoc xerado.

Marco de control

O marco de control contén as clases base e interfaces que definen o que significa ser un módulo ou un elemento básico de SimpleNLG. Ter unha definición de elemento básica garante que todos os módulos teñan as mesmas expectativas sobre como son os datos, mentres que ter unha definición base dun módulo permítelles executar outros.

As clases e interfaces definidas atópanse no paquete *simplenlg.framework*.

NLGElement

NLGElement é unha clase abstracta que representa a todos os compoñentes da linguaxe. Isto abarca tanto os compoñentes sintácticos como os estruturais dos documentos. Dous exemplos son a frase nominal (sintaxe) e o parágrafo (estrutura do documento). Un NLGElement pode referirse a outros elementos en forma de árbore. Así, un elemento do documento de nivel superior pode conter unha serie de seccións, que á súa vez conducen máis abaixo na árbore a parágrafos, oracións, varios tipos de frases e, finalmente, ás propias palabras.

Atributos

TIPO	NOME	DEFINICIÓN
String	baseForm	Para palabras, representa a forma base da palabra utilizada antes de que fora modificada por algún módulo.
ElementCategory	category	Define o tipo de NLGElement que é.
List<NLGElement>	complements	Estes son os compoñentes fillo deste elemento. Os complementos unidos forman este tipo de elementos. Por exemplo, nunha frase os complementos poderían representar as palabras individuais.
HashMap<String, Object>	features	Unha lista das características relevantes para este NLGElement. Non pode haber dúas características co mesmo nome, sen embargo, o valor pode ser unha colección ou unha matriz.
HashSet<String>	flags	Os atributos booleanos poderían agregarse ao mapa de características como se mencionou anteriormente. Sen embargo, este campo ten unha forma máis convinte de almacenar os valores booleanos. Só os indicadores que son true son almacenados no conxunto.
NLGElement	head	Usado principalmente nas frases, representa o elemento principal.
NLGElement	parent	Unha referencia ao nodo pai na <i>árbore</i> .
String	realisation	A realización actual do elemento. Noutras palabras, a representación textual do elemento.

Métodos

SAÍDA	NOME	PARÁMETROS	DEFINICIÓN
void	addComplement	NLGElement	Engade un elemento <i>fillo</i> a este elemento.
void	clearAllComplements		Elimina todos os elementos <i>fillo</i> deste elemento.

void	clearAllFeatures		Elimina todas as características deste elemento.
void	clearAllFlags		Elimina todos os flags deste elemento, establecendo os flags a false .
String	getBaseForm		Devolve a forma base da palabra.
ElementCategory	getCategory		Devolve o tipo deste elemento.
List<NLGElement>	getComplements		Devolve os elementos <i>fillo</i> deste elemento.
Object	getFeature	String	Devolve o valor da característica como un obxecto.
NLGElement	getFeatureAsElement	String	Devolve o valor da característica como un NLGElement.
String	getFeatureAsString	String	Devolve o valor da característica como unha cadea de texto.
NLGElement	getHead		Devolve o elemento principal deste elemento.
NLGElement	getParent		Devolve o elemento pai deste elemento.
String	getRealisation		Devolve a realización actual deste elemento.
boolean	hasFeature	String	Determina se este elemento ten un valor asociado a unha característica.
boolean	isA	ElementCategory	Determina se este elemento é do tipo indicado.
boolean	isFlagTrue	String	Determina se o flag indicado é true .
void	removeComplement	NLGElement	Elimina o <i>fillo</i> indicado deste elemento.
void	removeFeature	String	Elimina a característica indicada deste elemento.
void	removeFlag	String	Elimina o flag indicado deste elemento.
void	setBaseForm	String	Establece a forma base deste elemento.
void	setCategory	ElementCategory	Establece o tipo deste elemento.
void	setComplement	NLGElement	Elimina todos os elementos <i>fillo</i> existentes e engade só a este <i>fillo</i> .
void	setFeature	String, Object	Establece a característica indicada deste elemento co obxecto dado.
void	setFlag	String, boolean	Establece o flag indicado deste elemento co valor booleano dado.
void	setHead	NLGElement	Establece o elemento principal deste elemento.
void	setParent	NLGElement	Establece o elemento pai deste elemento.
void	setRealisation	NLGElement	Establece a realización actual deste elemento.

NLGModule

NLGModule é unha clase abstracta que define un módulo individual do framework. Todos os módulos implementados deben estender esta clase. É posible enlazar os módulos mediante a adición de módulos a outros existentes. Os módulos engadidos executaranse antes do propio módulo principal. Os módulos pódense agregar en función da prioridade, onde o módulo co valor máis baixo execútase antes que calquera outro.

Atributos

TIPO	NOME	DEFINICIÓN
static final String	SALIENCE_MODIFER	O modificador predeterminado para os valores de prioridade empregados principalmente ao agregar novos módulos ao comezo ou ao final da lista de módulos actuais. O valor é 100.
TreeMap<Integer, NLGModule>	modules	A lista de módulos que se deben executar antes de executar este módulo.

Métodos

SAÍDA	NOME	PARÁMETROS	DEFINICIÓN
int	addModule	NLGModule	Engade un módulo ao final da árbore dos módulos actuais. Devolve o valor de prioridade aplicado.
NLGModule	addModule	NLGModule, int	Engade un módulo na árbore no punto indicado pola súa prioridade. Se a árbore xa contén un módulo con esa prioridade, sobrescríbese e devolve o módulo antigo.
void	clearModules		Elimina todos os módulos da árbore.
Collection<NLGModule>	getModules		Devolve a colección de módulos asociados con esta orde de prioridade.
abstract void	initialise		Permite ao módulo realizar algunha operación antes de executarse.
abstract String	realise	List<NLGElement>	Fío de procesamento principal do módulo que realiza as accións da lista de NLGElements dada.
abstract String	realise	NLGElement	Fío de procesamento principal do módulo que realiza as accións do NLGElement dado.
NLGModule	removeModule	int	Elimina o módulo coa prioridade indicada na árbore. Devolve o módulo eliminado.
int	removeModule	NLGModule	Elimina o módulo indicado na árbore de módulos. Devolve a prioridade do módulo eliminado. Se non existira ese módulo, devólvese Integer.MIN_VALUE.

ElementCategory

ElementCategory é unha interface que debe ser implementada polas clases ou tipos numerados utilizados para definir as categorías sintácticas ou estruturais. Ter esta interface permite empregar varios tipos numerados (como para ter unha separación entre a estrutura sintáctica e a estrutura do documento), asegurando ao mesmo tempo que os módulos teñan acceso xenérico a todos estes tipos.

Método

SAÍDA	NOME	PARÁMETROS	DEFINICIÓN
boolean	equalTo	Object	Determina se o obxecto indicado é igual a esta categoría.

Lexicón

O lexicón é unha base de datos de información sobre palabras. Os datos sobre cada palabra inclúen:

- Forma base (p. ex: “can”)
- Categoría sintáctica (p. ex. “substantivo”)
- Características. Isto inclúe:
 - Información morfolóxica (tipo de inflexión, formas irregulares)
 - Información sintáctica (complementos, posición adx, etc.)

Clases

Lexicón

Interface/clase xenérica que permite aos usuarios:

- Obter unha palabra, dada a súa forma base (e opcionalmente categoría sintáctica).
- Obter unha palabra dado o seu ID.
- Obter unha palabra dada unha forma flexionada (por exemplo, “cans” en lugar de “can”) e categoría sintáctica.

XMLLexicon

Clase para cargar palabras a partir dun arquivo XML. Todo o arquivo é lido no momento da carga.

Módulos

Módulo sintáctico

O módulo sintáctico é utilizado para construír a sintaxe do texto mediante a creación de diferentes tipos de frases, por exemplo, frase nominal, frase verbal, frase adxectiva. Facer unha chamada de realización sobre estes elementos devolverá a forma base de todos os elementos.

Ademais de ser unha extensión da clase NLGModule, o procesador sintáctico tamén contén unha serie de clase de soporte para definir a linguaxe, como unha serie de tipos numerados para representar características particulares (xénero, aspecto, tempo). Todos eles aparecerán baixo `nlg.syntax.english`. Debería ser posible crear sintaxe adicional para outros idiomas.

Módulo ortográfico

O módulo ortográfico toma o texto básico e engade a puntuación e as maiúsculas. Por exemplo, “o mensaxeiro entregou a bicicleta verde a María” converteríase en “O mensaxeiro entregou a bicicleta verde a María.”. A primeira letra da oración ponse en maiúsculas e a oración termina cun punto. Estas accións realízanse ao chamar ao método de realización.

Módulo morfolóxico

O módulo morfolóxico controla a forma correcta das palabras. Encárgase da tarefa de pluralizar e do cambio de tempo entre as palabras. Por exemplo, a oración “María persegue a Carlos.” Está en tempo presente, se queremos cambialo a tempo pasado, o módulo de morfoloxía convertería “persegue” en “perseguiu” polo que a frase converteríase en “María perseguiu a Carlos.”.

Módulo de agregación

O módulo de agregación une as oracións, usualmente mediante o uso de ‘e’, ‘ou’ e ‘pero’.

Módulo Realizador predeterminado

O módulo realizador predeterminado enlaza os módulos antes mencionados. Un usuario pode construír os seus elementos e executar o realizador que, á súa vez, chamará a cada módulo para producir a cadea final completamente realizada.

Módulos de formateo

Hai dous tipos de módulos de formateo no paquete SimpleNLG. Un dos módulos engade as etiquetas HTML ao texto, por exemplo etiquetas de parágrafo “<p>” e listas “”. O segundo módulo realiza un formateo de texto simple, incluíndo o seguinte:

- Engadir o título do documento ao comezo do texto.
- Engadir títulos de sección nos lugares axeitados.
- Agregar saltos de liña para facilitar a lectura.
- Agregar elementos de lista con ‘*’.
- Agregar números para listas numeradas (por exemplo, “1.1 -”, “1.2 -”, etc.).