Extracción y aprovechamiento de DTDs emparejadas en corpus paralelos

Arantza Casillas
Departamento de Automática, Universidad de Alcalá
e-mail:arantza@aut.alcala.es

Joseba Abaitua
Facultad de Filosofía y Letras, Universidad de Deusto
e-mail:abaitua@fil.deusto.es

Raquel Martínez
Departamento de Sis. Informáticos y Programación, Facultad de Matemáticas
Universidad Complutense de Madrid
e-mail:raquel@eucmos.sim.ucm.es

Resumen

El artículo presenta un algoritmo que permite abstraer DTDs a partir de textos etiquetados en SGML. Estas DTDs se utilizan luego para generar textos similares. El algoritmo se ha aplicado a un corpus paralelo y con ello se han obtenido DTDs emparejadas que permiten generar nuevos documentos bilingües. Esta metodología supone una importante contribución al campo de la edición plurilingüe.

1 Introducción

Los lenguajes de etiquetado descriptivo, como SGML (*Standard Generalized Markup*) [ISO 86],[Barron 89], [Herwijnen 92], ofrecen interesantes posibilidades para agilizar los tradicionalmente laboriosos procesos de creación y mantenimiento de documentación bilingüe. Un corpus paralelo etiquetado con pautas normalizadas de utilización de SGML, como son las directrices TEI (*Text Encoding Initiative*), presenta un atractivo campo de estudio para ensayar técnicas de identificación automatizada de unidades estructurales y de gestión integral de documentación bilingüe. Un documento SGML debe estructurarse de acuerdo a una DTD (*Document Type Definition*) que define, a modo de gramática, la aparición de los elementos lógicos en el documento. La DTD especifica los elementos estructurales que pueden aparecer y su orden dentro del documento. Los componentes básicos de una DTD son las declaraciones de elemento, atributo y entidad (elementos como párrafos, oraciones, términos, etc; atributos como los identificadores únicos de los elementos; entidades como las de caracteres especiales, figura 1). Una DTD se utiliza para crear documentación estructurada o comprobar si un documento etiquetado cumple las reglas estructurales especificadas. El proceso habitual es crear el documento etiquetado a partir de una DTD predefinida, es decir, primero es la DTD y luego el documento SGML. Ello entraña el paso previo y nada trivial de selección del modelo de DTD que se ajuste mejor a la estructura lógica de la documentación que se quiere generar. Existen dos alternativas, adoptar una DTD conocida, de uso normalizado o estándar, o crear una a medida. Esta última opción es más costosa, pero tiene la ventaja de que la definición del tipo de documento puede ajustarse con

mayor precisión a las necesidades concretas. En ocasiones se puede dar el caso de que se disponga de un corpus etiquetado sin que en el proceso de asignación de etiquetas se haya tenido en cuenta ninguna DTD predefinida. Esto sucede con frecuencia en el campo de la lingüística de corpus, cuando se aplican técnicas de etiquetado automático a colecciones documentales disponibles en formato electrónico pero carentes de etiquetado descriptivo [Martinez 98b]. Presentamos en la segunda sección el procedimiento seguido para detectar y extraer DTDs a partir de documentos etiquetados en SGML. Esta labor se puede realizar manualmente, examinando uno a uno todos los documentos y deduciendo sus correspondientes DTDs, o automáticamente, mecanizando dicho proceso manual. La extracción automática presenta obvias ventajas como se explica más adelante. En la tercera sección mostramos cómo se pueden crear documentos bilingües aprovechando las DTDs extraídas del corpus paralelo. Para ello es necesario crear DTDs emparejadas que se utilizan en el proceso de generación del documento origen y meta. Mediante una estadística se mide el tanto por ciento de documento que genera la DTD por este procedimiento. Finalmente, se citan cuáles son las líneas que vamos a seguir en nuestro trabajo futuro y las conclusiones a las que hemos llegado.

2 Síntesis de DTDs

En ocasiones se dispone de un conjunto de documentos que están etiquetados, pero no se conoce cuál es la DTD que se ha seguido para marcarlos. La creación manual de una DTD es un proceso complejo, sobre todo cuando el número de documentos que se quiere abarcar es elevado. Por ello son muy útiles las herramientas que permiten obtener DTDs de manera automática sobre la base de documentos etiquetados.

2.1 Antecedentes

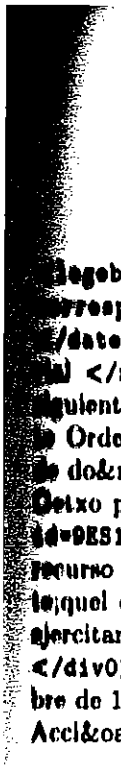
[Shafer 93] y [Ahonen 95] han estudiado la generación automática de DTDs. La idea básica del método empleado por [Ahonen 95], quien parte de las propuestas previas de [Sunniva 94], es formar un autómata finito determinista para cada elemento SGML, a partir de las instancias deducidas del documento etiquetado. Después generaliza cada uno de los autómatas y los convierte en expresiones regulares. Dichas expresiones se pueden transformar fácilmente en modelos de contenido SGML y de esta forma crear la DTD. Una técnica más sencilla pero igualmente eficaz es la utilizada por [Shafer 93]. Shafer resuelve el problema mediante la creación de una gramática generalizada que parte de las reglas estructurales que se deducen de los documentos etiquetados. Dichas reglas se combinan, generalizan y simplifican dando como resultado una gramática que, expresada mediante la correspondiente notación SGML se convierte en una DTD. El algoritmo de síntesis de DTDs que hemos diseñado está basado en la técnica empleada por Shafer. Nosotros no empleamos todas sus simplificaciones, pero hemos cambiado algunas e incorporado otras nuevas. La metodología aplicada en nuestro algoritmo se sirve de las técnicas de síntesis de gramáticas de la teoría de computación.[Cohen 91] define una gramática formal como un modelo matemático que permite especificar lenguajes, entendidos estos como conjuntos de cadenas. Podemos generar cadenas de caracteres a partir de una gramática o, a la inversa, deducir una gramática desde un conjunto finito de cadenas. En nuestro caso las cadenas se refieren a las unidades documentales y las DTDs son sus gramáticas. Siendo posible realizar dos tipos de operaciones sobre una misma DTD, derivación y síntesis, mediante la derivación pueden crearse documentos que aún teniendo una estructura lógica diferente comparten una misma gramática y, a la inversa, la síntesis permite derivar una única DTD a partir de diferentes documentos.

2.2 Fases del algoritmo

Cuando la estructura de un documento se expresa a través de una gramática, la definición de cada elemento se corresponde con una regla gramatical. Para escribir las reglas de la gramática vamos a utilizar la siguiente notación:

$$ELEMENTO \rightarrow DEFINICION$$

La definición de elemento puede incluir operadores de conexión AND(&), OR(|), concatenación (,) y



<legebi> <text> <body> <opener> <seg31 id=13ES1 corresp=13EU1> Mediante </seg31> <title id=ctES1 corresp=ctEU1> Orden Foral </title> <num> n&uacuate;mero 3607/94, </num> <date> 9 de Noviembre de 1994 </date> <name id=nmES1 corresp=nmEU1> del Diputado Foral de Medio Ambiente y Acci&oacuate;n Territorial </name> <seg3 id=13ES1 corresp=13EU1> ha adoptado la resoluci&oacuate;n cuya parte dispositiva es la siguiente: </seg3> </opener> <div0> <div1> Primero: Revocar el nombramiento provisional otorgado mediante Orden Foral n&uacuate;mero <num num=60394> 603/94 </num>, de <date> 22 de febrero </date>, a favor de doña Ana Fern&aacuate;ndez Gutierrez-Crespo para el puesto de Tesorer&iacuate;a del Ayuntamiento de Getxo por incapacidad laboral transitoria de su titular, por haber fallecido este &uacuate;ltimo. </div1> <seg9 id=9ES1 corresp=9EU1> La anterior resoluci&oacuate;n es definitiva, contra la misma podr&aacuate; interponerse recurso contencioso administrativo en el plazo de dos meses contados desde el d&iacuate;a siguiente a &aacuate;quel en que tenga lugar la notificaci&oacuate;n del presente escrito sin perjuicio de que los interesados puedan ejercitar cualquier otro recurso que estimen pertinente de acuerdo con la legislaci&oacuate;n vigente. </seg9> </div0> <closer> <dateline> <rs type=place id=PES1 corresp=PEU1> Bilbao </rs>, <date> 9 de noviembre de 1994 </date>. </dateline> <name id=dcES1 corresp=dcEU1> El Director General de Medio Ambiente y Acci&oacuate;n Territorial , Ander Salaberria Amesti </name> </closer> </body> </text> </legebi>

Figura 1: texto etiquetado en castellano

los paréntesis para establecer prioridades. Además cada elemento que se incluye en la definición puede ir acompañado de los siguientes indicadores de ocurrencia: opcional (?), repetición y obligatorio (+), repetición y opcional (*).

Las operaciones de generalización y reducción se presentan de la forma:

$$REGLA(S)_{ORIGINAL(ES)} \Rightarrow$$

$$REGLA(S)_{REDUCIDA(S)}$$

donde la parte izquierda representa el conjunto de reglas originales y la parte derecha el resultado de la operación. Las reglas tendrán la estructura que se ha indicado anteriormente. Los principales pasos que seguimos, al igual que Shafer, para abstraer DTDs a partir de documentos etiquetados son: extracción, generalización, reducción de reglas gramaticales y codificación de la DTD. A continuación vamos a explicar en detalle cada uno de los pasos del algoritmo y comentar las diferencias que se dan con el de Shafer:

- **Extracción de reglas, atributos y entidades:** Se analiza el texto etiquetado para deducir las reglas gramaticales que conforman la estructura lógica del documento. Al mismo tiempo, se extraen los atributos contenidos en los elementos. También se comprueba si existe alguna entidad en el texto. En este paso Shafer sólo realiza la combinación de reglas y extracción de atributos sin tomar en cuenta las entidades. En la figura 2 se muestran las reglas extraídas del texto etiquetado que aparece en la figura 1.

- **Abstracción de cada una de las reglas extraídas:** Una vez que se han deducido las definiciones de todos los elementos se deben generalizar y reducir para crear una gramática que represente con precisión el conjunto de documentos de partida. Es decir, si no generalizamos ni reducimos las reglas, la gramática resultante únicamente representará a los documentos de partida y no será posible deducir de ella documentos diferentes a los iniciales. Por ejemplo, si tenemos las reglas:

$$A \rightarrow b, b$$

$$A \rightarrow b, b, b$$

y no las generalizamos con la gramática resultante, no podremos generar un documento de la forma:

$$A \rightarrow b, b, b, b$$

Con la regla resultante de la generalización de las dos reglas primeras sí se podrá generar:

$$A \rightarrow b, b; A \rightarrow b, b, b \Rightarrow A \rightarrow b+$$

La generalización también se puede aplicar a grupos de elementos, lo que Shafer no hace, como en el ejemplo que se expone a continuación:

$$A \rightarrow b, c, d, b, c, d, b, c, d \Rightarrow A \rightarrow (b, c, d)+$$

- **Reducción de reglas:** Al igual que Shafer, el objetivo de este paso es compactar la gramática. Tras los pasos de extracción y generalización puede ocurrir que varias reglas gramaticales difieran en la manera de derivar un mismo elemento. Esta situación plantea el inconveniente de que la gramática será imprecisa ya que no se sabrá con certeza qué regla se debe aplicar. Para evitarlo debemos simplificar y agrupar todas las definiciones de un mismo elemento en una única regla gramatical. Mientras se puedan hacer simplificaciones, el proceso de reducción que se aplica por defecto funciona recursivamente, de acuerdo al siguiente orden preestablecido:

  1. **Simplificación ?:** Se realiza sobre dos reglas con igual parte izquierda y derecha, salvo en un elemento de la parte derecha, que está presente en una y ausente en la otra:

     $$A \rightarrow b, c, d; A \rightarrow b, d \Rightarrow$$

     $$A \rightarrow b, c?, d$$

     O sobre una misma regla cuando el contenido del elemento está formado por varias cadenas iguales excepto una, que puede contener elementos adicionales. Esto se ve en los siguientes casos:

     $$A \rightarrow b, c, d, e, b, e, b, e \Rightarrow$$

     $$A \rightarrow (b, (c, d)?, e)+$$

     $$A \rightarrow b, e, b, d, e, b, e, b, e \Rightarrow$$

     $$A \rightarrow (b, d?, e)+$$

     Esta reducción no es tenida en cuenta por Shafer.

  2. **Simplificación *:** Es similar a la simplificación ?. Se aplica sobre elementos con el indicador de ocurrencia +. Para dos reglas:

     $$A \rightarrow (b, c)+, d; A \rightarrow d \Rightarrow$$

     $$A \rightarrow (b, c)*, d$$

     Para una regla:

     $$A \rightarrow b, (c, d)+, e, b, e, b, e \Rightarrow$$

     $$A \rightarrow (b, (c, d)*, e)+$$

     $$A \rightarrow b, d+, e, b, e, b, e \Rightarrow$$

     $$A \rightarrow (b, d*, e)+$$

  3. **Simplificación |:** Se aplica en tres situaciones. La primera, que Shafer no considera, es:

     $$A \rightarrow b, c, d, b, c, e, b, c, d, b, c, d \Rightarrow$$

     $$A \rightarrow (b, c, (d|e))+$$

     La segunda, se aplica sobre dos reglas:

     $$A \rightarrow b, c, d; A \rightarrow b, e, f, d \Rightarrow$$

     $$A \rightarrow b, (c|(e, f)), d$$

     Por último, se aplica para unir todos los posibles contenidos de las todas las reglas con igual parte izquierda:

     $$A \rightarrow b, c; A \rightarrow e, d \Rightarrow A \rightarrow (b, c) \mid (e, d)$$

  4. **Simplificación &:** Se aplica a reglas con igual parte izquierda y cuya parte derecha esté formada por los mismos elementos en cualquier orden. Se puede aplicar a dos reglas:

     $$A \rightarrow b, c, d, e; A \rightarrow e, d, c, b \Rightarrow$$

     $$A- > b\&c\&d\&e$$

     O sobre una única regla (no implementada por Shafer):

     $$A \rightarrow b, c, d, e, d, e, c, b \Rightarrow$$

     $$A \rightarrow (b\&c\&d\&e)+$$

5. **Simplificación con #PCDATA**: a toda regla que contiene #PCDATA se le aplica la reducción:

$$A \rightarrow b, \#PCDATA \Rightarrow$$

$$A \rightarrow (b|\#PCDATA)+$$

6. **Eliminación de reglas idénticas**: Es decir, todas las reglas cuya parte derecha e izquierda coincide en todo se reducen a una:

$$A \rightarrow b\&c; A \rightarrow b\&c; A \rightarrow b\&c \Rightarrow$$

$$A \rightarrow b\&c$$

Cuando la reducción se puede aplicar sobre una o más reglas, el algoritmo intenta reducir primero cada regla de manera individual y luego por pares. Este algoritmo permite establecer un orden distinto del previsto por defecto para realizar las simplificaciones e incluso especificar cuáles simplificaciones se quiere aplicar. Por este motivo, primero es necesario comprobar la compatibilidad de las reglas que se desea simplificar, ya que no siempre se podrán llevar a cabo, como en el caso siguiente:

$$A \rightarrow b, c+$$

$$A \rightarrow b$$

Con estas dos reglas no es posible aplicar la simplificación ?, ya que se produce una incompatibilidad semántica entre los indicadores de ocurrencia. El operador ? solamente se puede aplicar cuando el elemento aparece una vez o ninguna, pero no cuando puede no aparecer o aparecer varias veces. En este caso lo correcto sería aplicar el operador *.

- **Codificación de la DTD**: Las reglas obtenidas junto con los atributos extraídos para cada elemento y las entidades detectadas, debidamente codificados según la sintaxis SGML, darán lugar a una DTD. Se considera también la posibilidad de que las etiquetas de un elemento se puedan o no simplificar. La codificación de la DTD se muestra en las figura 3.

3 Aplicación del algoritmo a un corpus paralelo

La síntesis de DTDs resulta útil cuando se dispone de un conjunto de documentos etiquetados y queremos conocer la DTD en la que están basados, o bien la DTD que les pueda corresponder. Es decir, este algoritmo de síntesis de DTDs también se puede aplicar en tareas como la asignación de DTDs a documentos bilingües etiquetados en SGML.

3.1 El corpus paralelo

El corpus paralelo que ha servido de muestra para este experimento está formado por una colección de documentos bilingües jurídico-administrativos en castellano y euskara procedentes del Boletín Oficial de Bizkaia (BOB), años 1990-1995. El tamaño total del corpus original es de 7 millones de palabras en cada idioma, pero, debido a diversos problemas de formato, asimetrías de contenido y otra clase de ruidos, para el experimento se ha extraído una muestra reducida de 500.000 palabras en cada idioma. El subcorpus seleccionado ha sido anotado en SGML de acuerdo con las directrices TEI [Sperberg 94] y [Ide 94], mediante etiquetadores que identifican los principales elementos lógicos de cada documento. A su vez estos etiquetadores establecen correspondencias entre los elementos equivalentes en cada versión del documento, como se ha descrito en [Martinez 98b], [Martinez 98a].
El subcorpus está formado únicamente por *órdenes forales* que constituyen el tipo de documento jurídico-administrativo más numeroso del BOB. La figura 1 contiene una muestra de una Órden Foral en castellano etiquetada.

3.2 Asignación de DTDs al subcorpus del BOB

El algoritmo expuesto en la sección 2 se ha aplicado al subcorpus, de esta manera se ha obtenido el conjunto de DTDs que refleja la estructura lógica de las órdenes forales del BOB en los dos idiomas. El algoritmo se ha aplicado de la siguiente manera: Primero se han extraído las DTDs de las órdenes forales de la

$legebi \rightarrow text$
$text \rightarrow body$
$body \rightarrow opener, div0, closer$
$opener \rightarrow seg31, title, num, date, name, seg3$
$title \rightarrow \#PCDATA$
$num \rightarrow \#PCDATA$
$date \rightarrow \#PCDATA$
$seg31 \rightarrow \#PCDATA, rs, \#PCDATA$
$seg3 \rightarrow \#PCDATA, rs, \#PCDATA$
$name \rightarrow \#PCDATA, rs, \#PCDATA$
$div0 \rightarrow div1, seg9$
$div1 \rightarrow \#PCDATA, num, \#PCDATA, date,$
$\#PCDATA$
$num \rightarrow \#PCDATA$
$date \rightarrow \#PCDATA$
$seg9 \rightarrow \#PCDATA, num, \#PCDATA, num,$
$\#PCDATA, date, \#PCDATA, num,$
$\#PCDATA, date, \#PCDATA, num,$
$\#PCDATA, date, \#PCDATA, num,$
$\#PCDATA, num, \#PCDATA, date,$
$\#PCDATA$
$num \rightarrow \#PCDATA$
$num \rightarrow \#PCDATA$
$date \rightarrow \#PCDATA$
$num \rightarrow \#PCDATA$
$date \rightarrow \#PCDATA$
$num \rightarrow \#PCDATA$
$date \rightarrow \#PCDATA$
$num \rightarrow \#PCDATA$
$num \rightarrow \#PCDATA$
$date \rightarrow \#PCDATA$
$closer \rightarrow dateline, name$
$dateline \rightarrow rs, date$
$name \rightarrow rs, rs$
$rs \rightarrow \#PCDATA$
$date \rightarrow \#PCDATA$

Figura 2: reglas extraidas del texto en castellano

```
<!ELEMENT LEGE - - (TEXT)>
<!ELEMENT TEXT - - (BODY)>
<!ELEMENT BODY - - (OPENER, DIV0, CLOSER)>
<!ELEMENT OPENER - - (SEG31, TITLE, NUM,
DATE, NAME? SEG3)>
<!ELEMENT (SEG3,SEG31) - - (#PCDATA))>
<!ELEMENT (DIV0) - - (DIV1, SEG9?, SEG10?)>
<!ELEMENT (DIV1) - - (#PCDATA|RS|DATE|NUM)+>
<!ELEMENT (SEG10,) - - (#PCDATA|RS|NUM)+>
<!ELEMENT (CLOSER) - - (DATE?, NAME?)>
<!ELEMENT (NAME) - - (RS)+>
<!ELEMENT (TITLE, NUM, DATE, RS, SEG9)
- -(#PCDATA)>
<!ATTLIST RS TYPE (ORGANIZATION| LAW| PLACE|
UNCAT) #IMPLIED>
<!ATTLIST SEG9 CORRESP #IMPLIED>
<!ATTLIST RS CORRESP #IMPLIED>
<!ATTLIST SEG31 CORRESP #IMPLIED>
<!ATTLIST SEG3 CORRESP #IMPLIED>
<!ATTLIST TITLE CORRESP #IMPLIED>
<!ATTLIST NAME CORRESP #IMPLIED>
<!ATTLIST SEG10 CORRESP #IMPLIED>
<!ATTLIST SEG9 ID #IMPLIED>
<!ATTLIST RS ID #IMPLIED>
<!ATTLIST SEG31 ID #IMPLIED>
<!ATTLIST SEG3 ID #IMPLIED>
<!ATTLIST TITLE ID #IMPLIED>
<!ATTLIST NAME ID #IMPLIED>
<!ATTLIST SEG10 ID #IMPLIED>
```

Figura 3: DTD para textos en castellano

gua origen y de la lengua meta. Después, se establecen las correspondencias entre las DTDs de las denes del mismo tipo en los dos idiomas mediante na tabla general de emparejamiento de DTDs diñada al efecto. Esta tabla de emparejamientos se educe manualente comparando las DTDs obtenidas. as DTDs obtenidas mediante el algoritmo de abstracción refleja la estructura de todos los documentos imilares al que se muestra en las figura 1. Por ello la DTD que se muestra en las figura 3 no se corresponde exactamente con la estructura lógica del documento de la figura 1, sino que son una síntesis de todos los documentos de este mismo tipo. La figura 2 ilustra las reglas extraídas del documento en castellano de la figura 1 que junto con las reglas extraidas de otros documentos del mismo tipo dará lugar a la DTD de la figura 3 una vez aplicado el algoritmo de simplificación. La estructura particular de cada documento se deriva de una estructura general que es la que se refleja en la DTD ([Akpotsui 92]), elgiendo en la DTD los elementos opcionales que le correspondan. Los elementos opcionales <seg9>, <seg10>, etc. de la figura 3 son los que determinarán la estructura particular del documento de la figura 1.

4 Evaluación del algoritmo

El subcorpus del BOB ha servido para probar el algoritmo de síntesis de DTDs. De este estudio se ha extraído una estadística que muestra el porcentaje (medido en palabras) de la estructura del documento (origen y meta) que una DTD puede asignar (ver tabla 1). Para completar la estructura del documento, será necesario seleccionar los elementos opcionales que son particulares del docuemnto en cuestión. Sobre un total de 1014 documentos analizados el mejor resultado se ha obtenido en un documento de menos de 500 palabras con el 67.28% de estructura asignada. El peor resultado se ha obtenido con un documento de más de 1000 palabras con un resultado de 0.16% de documento asignado.

5 Aprovechamiento de DTDs emparejadas en la edición de documentos bilinges

El algoritmo desarrollado tiene aplicación directa en los procesos de edición de documentación bilingüe. Una vez que se han obtenido DTDs para los distintos tipos de documentos en ambos lados del corpus paralelo, se ha procedido a relacionar las DTDs de los documentos equivalentes en las dos lenguas. El resultado es un conjunto de DTDs emparejadas. Se denomina DTD emparejada al par de DTDs que describe para cada lengua la estructura lógica de un tipo de documento. La composición estructural de un documento en un entorno de edición basado en SGML está guiada por la DTD. La DTD determina los elementos obligatorios e indica los elementos opcionales que deberán ser elegidos por el usuario para configurar la estructura particular del documento. Un entorno de edición bilingüe basado en SGML puede aprvechar las DTDs emparejadas para deducir en dos pasos secuenciales la estructura primero del documento origen y posteriormente la del documento meta. Como aplicación del algoritmo de extracción de DTDs se está desarrollando un prototipo de editor bilingüe, cuyos fundamentos metodológicos se describen a continuación. El interfaz gráfico de este entorno se está desarrollando en Itcl/Tk e integra un conjunto de algoritmos implementados en C++ que son los que se encargan de guiar al usuario en la confección de la estructura particular del documento origen y meta.

5.1 Derivación del documento de origen

En la fase de creación del documento origien, el entorno ofrece las funciones típicas de un sistema de edición SGML de propósito general. Mediante menús, el usuario podrá elegir entre crear el documento origen a partir de una DTD conocida por el entorno o crear una DTD nueva. Cuando la DTD no es conocida, el entorno aplicará un procedimiento de validación de la DTD.
Una vez validada la DTD, se creará una representación interna de la DTD consitente en un grafo y dos

Número de palabras	Número de documentos	% de documento generado
0-500	918	29.5
500-1.000	52	11.07
Más de 1000	44	2.71

Tabla 1: % de documento generado por la DTD

tablas. El grafo representa las relaciones jerárquicas entre los distintos elementos de la DTD y las dos tablas contienen los valores de las entidades y los atributos para cada elemento lógico del documento. A partir de este grafo se creará una plantilla inicial del documento formada por los elementos obligatorios. La plantilla indicará las partes en que se pueden incluir elementos opcionales y las diferentes posibilidades. Además mostrará al usuario los posibles valores de los atributos asignados a los elementos y las entidades que se pueden incluir.
Para determinar si un elemento lleva atributos o no se examinará la tabla de atributos. Las entidades que pueden aparecer en el documento estarán registradas en la tabla de entidades. Los datos representados en el grafo se utilizan también para verificar si una etiqueta puede ser simplificada, qué elementos pueden ser suprimidos, o si es posible cambiar los valores asignados por defecto a los atributos y para crear, borrar y manipular entidades.

5.2 Derivación del documento meta

Una vez conocida la DTD de origen, el primer paso para crear la estructura del documento meta consiste en localizar, por medio de la tabla general de emparejamiento, la DTD meta. A partir de aqui se podrá generar la estructura particular del documento meta. Para ello será necesario tener resuelta la estructura particular del documento de origen que, junto con la estructura general contenida en la DTD extraída de la tabla general de emparejamiento, proporcinan los elementos necesarios para deducir la estructura particular del documento meta. En el caso de que el documento de origen se haya creado a partir de una DTD nueva, la tabla general de emparejamiento no contendrá la DTD meta correspondiente, por lo cual el usuario deberá crear una DTD meta, de manera análoga a como creó la DTD de origen, que deberá ser validada por el sistema.

En el estado actual de desarrollo del entorno de edición, únicamente se resuelven las estructuras particulares de los documentos bilingües y la tabla general de emparejamientos únicamente contiene las DTDs del subcorpus BOB. Pero la metodología puede aplicarse a otras colecciones de documentos bilingües.

5.3 Ventajas de la extracción automática de DTDs

En esta sección hemos comprobado la utilidad del algoritmo de extracción de DTDs. Las principales ventajas se pueden resumir en los siguientes puntos:

- Cuando no se dispone de DTDs estándar, el algoritmo permite deducir automáticamente la DTD que describe la estructura para cada tipo de documento en un corpus estiquetado.
- El proceso manual de creación de DTDs, que además de ser muy costoso es poco fiable, es reemplazado por un proceso rápido y sistemático.
- A mayor número de muestras de documento, la extracción de la DTD será más precisa. Con todo, el algoritmo tiene un buen comportamiento con un número reducido de muestras.
- Dado que la extracción automática de DTDs reduce el coste de creación, la adecuación de la DTD a las necesidades concretas del usuario será más factible.

Por otro lado, el emparejamiento de DTDs

- Ayuda a crear la estructura del documento origen y a deducir la estructura del documento meta.

Es de gran utilidad en el tratamiento no sólo de documentación bilingüe sino también multilingüe. Es suficiente con disponer de tantas DTDs como idiomas se utilicen.

Trabajo futuro

El subcorpus de documentos bilingües que se ha utilizado para la sintesís automatizada de DTDs está alineado a distintos niveles, que van desde la terminología y nombres propios hasta unidades estructurales de los documentos, pasando por oraciones y párrafos enteros [Martinez 98b]. Nuestro objetivo es llegar a integrar estos elementos alineados en los recursos que maneje el entorno de edición bilingüe, de manera que el usuario no disponga solamente de una guía para generar la estructura particular de cada documento, sino que además pueda dotarla de contenidos. En la actualidad estamos probando un conjunto de algoritmos que vuelcan los segmentos alineados del subcorpus en una colección de bases de datos que permiten su extracción desde el entorno de edición.

7 Conclusiones

En este artículo hemos presentado un algoritmo para abstraer DTDs automáticamente a partir de un corpus de documentos etiquetados. Además se ha mostrado la aplicación del algoritmo a un corpus de documentos bilingües alineados. Como resultado se han obtenido DTDs emparejadas que a su vez son de gran utilidad para dirigir el proceso de generación de la estructura de los documentos origen y meta en un entorno de edición bilingüe.

Referencias

[Ahonen 95] Ahonen E. Automatic generation of SGML content models. *Electronic Publishing*, 8(2&3), 195-206, 1995.

[Akpotsui 92] Akpotsui E., Quint V. Type transformations in Structured Editing Systems. *Proceedings of Electronic Publishing*, 27-41, 1992.

[Barron 89] Barron D. Why use SGML.*Electronic Publishing*, 2(1), 3-24, 1989.

[Cohen 91] Cohen D. Intoduction to Computer Theory.*John Willey & Sons*, New York, 1991.

[Herwijnen 92] Van Herwijnen E. Practical SGML. *Kluwer academic publishers*, 1992.

[Ide 94] Ide N. and Véronis J. The Text Encoding Iniciative (TEI), P1. *TEI*, 1994.

[ISO 86] Information Processing - Text and Office Systems – Standard Generalized markup Language (SGML). *Intenational Organization for Standardization. Ref. No. ISO 8879:1986*, 1986.

[Lange 97] Langé J., Gaussier E., Daille B. Bricks and Skeletons: Some Ideas for the Near Future of MAHT. *Machine Translation*, 12, 39-51, 1997.

[Martinez 98a] Martínez R., Abaitua J. and Casillas A. Bitext Correspondences through Rich Mark-up. *36th Annual Meeting of the Association for Computational Linguistics (ACL'98)*, 812-818, 1998.

[Martinez 98b] Martínez R., Casillas A. and Abaitua J. Aligning tagged bitexts. *Sixth Workshop on Very Large Corpora*, 102-109, 1998.

[Shafer 93] Shafer K. SGML grammar structure. *Annuall Review of OCLC research*, 1993.

[Sperberg 94] Sperberg-McQuenn C. Burnard L. Guidelines for the Encoding and Interchange of Machine-Readable Texts, TEI document P3. *ACH-ACL-ALLC, Chicago, Illinois, USA and Oxford, England* 1994.

[Sunniva 94] Sunniva M., Solstrand S. Automastik generering av DTD fra SGML-kodet materiale. *M.Sc.thesis, Institutt for infromasjonsvitenskap, Universitetet i Bergen*, 1994.