ETIQUETADO DE ARTICULOS CIENTIFICOS: un primer intento de construcción de un sistema basado en el conocimiento para tratar ficheros ASCII para publicar con Ventura.

María José Aramburu Cabo[1]
Peter Hammersley[2]

[1]*Universitat Jaume I*
Campus Penyeta Roja
12071 Castellón
Tel.: 964/345775
Fax: 964/345847
España

[2]*Middlesex University*
The Burroughs, Hendon
London NW4 4BT
Inglaterra

RESUMEN

En este artículo presentamos el proceso de desarrollo de un sistema que etiqueta el texto ASCII de un artículo científico con las etiquetas aceptadas por el programa de autoedición Xerox Ventura Publisher. Esta labor, en la actualidad resuelta de forma manual, constituye un eslabón esencial en la automatización de la edición de una revista.

Su mayor novedad se centra en que es una implementación en PROLOG de un reconocedor inteligente de elementos textuales a través de su "comprensión" semántico-sintática de alto nivel. Las reglas utilizadas se basan en la gramática libre de contexto que representa la estructura de la clase de documentos que reconoce.

El sistema resultante trabaja exitosamente para los artículos que guardan la estructura establecida y, además, realiza una ordenación automática de las referencias que se hagan en el texto. Esto permite una edición automática del artículo, que acelera todos los procesos previos a su publicación. Un primer prototipo ha sido aplicado a artículos ASCII recibidos por e-mail en la editorial de la revista científica "The Computer Journal" obteniendose un texto en el que ya no es necesaria la intervención de un maquetador para su publicación.

Como posteriores desarrollos de esta investigación, actualmente se esta trabajando en la linea de crear un sistema basado en el conocimiento que admita más estructuras de documento, nuevas técnicas de marcado a mayor nivel de detalle y que ofrezca un interfaz al usuario que le permita definir el estilo y la forma del texto.

Palabras Clave: Sistema Basado en el Conocimiento, PROLOG, Reconocedor de Texto, Inserción de Etiquetas, Transmisión y Edición de Documentos, SGML, ODA (Office Document Architecture), Gestión de Referencias, Documentos Estructurados.

1. Introducción.

En el mundo académico y de la investigación, la publicación de prensa científica es de vital importancia, ya que supone la comunicación y el esparcimiento de los temas de estudio por todo el área de interés. La publicación de artículos científicos debe ser muy rápida, ya que cuanto antes esté disponible la información para otros científicos, antes serán posibles nuevos desarrollos.

La automatización del proceso de publicar un documento comenzó hace años con el uso de los procesadores de texto, pero la real incursión de la informática en el mundo de las publicaciones llegó con los sistemas de composición controlados por computador. Al mismo tiempo, los lenguajes de marcado de texto también evolucionaron para aprovechar las nuevas posibilidades ofrecidas por la tecnología informática.

Los dos eventos clave para la amplia aplicación actual de la informática a la industria de la publicación han sido [1] el espectacular descenso del costo de los ordenadores y su amplio uso en todos los campos, además de otros hechos como la mejora de los medios de comunicación electrónica y la adaptación de las tecnologías láser para la impresión de documentos con una alta calidad y rapidez.

Los sistemas de autoedición y la publicación electrónica de documentos, son dos nuevos conceptos que representan los últimos desarrollos en este área. Junto con ellos, otros avances en el software como han sido los lenguajes de descripción de páginas, han hecho que la publicación electrónica de documentos y la composición asistida por ordenador, sean unos importantes campos de aplicación de nuevas técnicas informáticas.

1.1 Propósito de este trabajo.

El desarrollo y la publicación de documentos e incluso su acceso por el lector, están evolucionando. Actualmente, el proceso del texto desde que es escrito por el autor hasta que es accesible por el lector, está siendo muy simplificado y acelerado. Esto se refleja en una mayor colaboración entre los participantes en la tarea de publicación.

En el caso concreto de los artículos científicos, se ha intentado asignar al autor la labor de dar forma al documento al mismo tiempo que lo escribe. Para ello, el redactor ha de facilitar al autor un código genérico de composición, lenguaje de marcado o procesador de texto con claras instrucciones y de fácil manejo. Sin embargo, la experiencia ha demostrado los inconvenientes de este método, y la necesidad de buscar nuevas alternativas debido, principalmente, a las dificultades que encuentran los autores para ajustarse a los códigos y procesadores suministrados y su falta de información sobre la apariencia final del texto [2].

Esto hizo pensar en la necesidad de un sistema automatizado que, en lugar del autor, insertara en el texto el código genérico propio del editor; es decir, un programa que diera forma al texto según su estructura específica. Esto evitaría la tediosa labor para el autor de adaptarse a diferentes revistas y eliminaría los errores usuales debidos a su inexperiencia en temas de publicación.

El sistema a desarrollar debía contener todo el conocimiento necesario para determinar la forma que se debe dar a una porción de texto, al tiempo que debía reconocer su estructura (del mismo modo en que lo haría un redactor o editor de documentos).

La solución al anterior reto que se explora en este trabajo, se basa en el desarrollo de un sistema para el caso particular de señalar texto ASCII según el método usado por el programa de autoedición Xerox Ventura Publisher [1]. El sistema resultante basado en técnicas de sistemas basados en el conocimiento, señala cada párrafo de texto con la etiqueta adecuada dependiendo de su posición dentro de la estructura del documento. De esta manera, la composición y el formato del texto podrían ser especificados e imprimidos de manera automática y con el mínimo esfuerzo por parte de autores, redactores y editores.

Además de lo anterior, el sistema ofrece la posibilidad de realizar la ordenación y numeración de las referencias insertadas en el texto automáticamente. Esto libera al autor de tan tediosa labor, y asegura la inexistencia de errores.

La implementación del sistema se realizó en PROLOG mediante la representación en forma de reglas de la gramática libre de contexto correspondiente a la estructura preestablecida para el documento.

En las secciones 2 y 3 describiremos el estado del arte, describiremos en la sección 4 el trabajo realizado, para acabar en la 5 estudiando sus limitaciones.

2. Sistemas de preparación de documentos.

2.1 Marcado de texto.

En el método tradicional de publicación, el marcado de texto consiste en las señales y anotaciones hechas por el diseñador o el redactor con el fin de indicar al compositor cual debe ser la apariencia del documento impreso. En el caso de los procesadores de texto, marcado se refiere a los códigos e instrucciones insertadas en un documento por el programa de que se trate.

Entre los lenguajes y métodos de marcado de texto se distinguen dos tipos:

- **Marcado de procedimiento.** En él se especifica a un bajo nivel de detalle el diseño de cada párrafo. Este método conduce fácilmente a errores, es inflexible a cambios de forma y es de difícil uso para no expertos.
- **Marcado descriptivo.** Se basa en los siguientes postulados que definen sus propiedades [3]:
    - El marcado debe definir la estructura del texto, más que sus características físicas.
    - El marcado debe ser entendido, sin ambigüedad, tanto por programas como por personas.

Con este método, la presentación del texto será determinada por el programa que interprete el marcado. Sus ventajas surgen de su carácter genérico.

Hay que señalar que aunque algunos autores consideran que esta forma de marcar permite separar la estructura de la apariencia del texto, esto no es posible ni deseable [2].

2.2 La evolución de los sistemas de preparación de documentos.

Al mismo tiempo que la tecnología y el costo del hardware y el software han ido evolucionando, los sistemas de preparación de documentos han tratado de balancear el esfuerzo de producción con la calidad del producto.

En los años setenta, los sistemas desarrollados utilizaban un marcado de procedimiento, con el inconveniente de necesitar ser ejecutados para comprobar los efectos especificados en los comandos. Los macroprocesadores permitian definir secuencias de comandos con otros de más fácil manejo (p.ej.: *ms, mm*).

En los años ochenta, se han desarrollado y utilizado extensamente el sistema TEX [4] y el conjunto de programas que lo soportan. Los comandos y parámetros proporcionados por TEX son muchos más que los de *Troff* [5], y además su sintaxis es más regular e incluye nociones para anidar así como de alcance de los comandos.

Una importante derivación de TEX ha sido su macroprocesador LATEX, que establece un modo "lógico" de uso y que, al mismo tiempo, permite un acceso fácil al subyacente TEX. Es necesario señalar, la importante innovación que LATEX supuso por su enfoque lógico del marcado de textos.

El avance más importante (realizado durante los años ochenta) en los sistemas de preparación de documentos, han sido las impresoras láser junto con los **lenguajes de descripción de**

páginas. Este método describe un objeto gráfico como un conjunto de líneas y curvas, e información de su localización en la página. PostScript [6] [7] se ha convertido en un estándar para impresoras láser e incluso para máquinas de composición.

Otro importante avance de los últimos años son los **sistemas de edición interactivos**. En ellos, el usuario interacciona con el sistema y ve los cambios en tiempo real sobre la pantalla (WYSIWYG: What You See Is What You Get). Sin embargo, estos sistemas no son tan potentes como los batch, son más tediosos de usar por los expertos y tienden a concentrarse más en los aspectos visuales que en la estructura lógica del documento [8].

En el futuro, lo deseable sería una combinación de sistemas batch e interactivos. Esto constituye un área de investigación sobre el que se está trabajando actualmente con mucha intensidad.

3. Estándares para documentos estructurados.

Pocos de los métodos actuales para transmisión electrónica de documentos aportan la posibilidad de enviar, junto con el texto, información de cómo el documento será presentado después de impreso[1] o lo que es lo equivalente, su estructura. Esta ausencia de información sobre la estructura del texto, puede causar una degeneración del mensaje transmitido por el autor.

De cara al futuro será necesario un método estándar para definir la estructura de un documento que permita al autor expresar su mensaje sin restricciones y al lector hacer uso de la información recibida de la mejor manera que considere oportuna.

3.1 S.G.M.L.

S.G.M.L. [3][9] (**Standard Generalized Markup Language**) fue diseñado para permitir añadir información estructural a un documento por medio de la inserción de secuencias de caracteres definidas por el usuario en la cadena del texto.

SGML no establece unas reglas de cómo codificar un documento, sino que formaliza un conjunto de reglas que permiten al autor definir cómo ha decidido codificar su documento.

La estructura de un documento SGML, se define formalmente al comienzo del mismo en un conjunto de declaraciones que especifican la clase del documento y que constituyen la **Definición del Tipo de Documento** (DTD). En la DTD se define el **Identificador Genérico** (etiqueta) de cada elemento del documento y el modelo de contenido que define qué subelementos y cadenas de caracteres son aceptables.

El procesamiento posterior depende de la aplicación del documento. Si va a ser impreso, el parser que analice su estructura pasará el control a un programa que sustituya cada etiqueta por el conjunto de comandos del subyacente sistema de composición. En otro caso, el control pasará a otro tipo de procedimiento. Normalmente, cada identificador genérico tendrá asociado un procedimiento distinto.

Entre las muchas ventajas que este lenguaje proporciona al proceso previo a la publicación destacan:

- Facilita el intercambio de documentos.
- El texto puede ser transmitido de forma electrónica y ser impreso con diferentes estilos en los diferentes estados de producción [10].
- Tanto editores como redactores ahorran tiempo y trabajo en el proceso de publicación de documentos.

---

[1] LaTeX es un notable ejemplo de la difusión de un sistema que permite enviar electrónicamente un formato. A pesar de que su manejo puede resultar complejo y sus presentaciones son rígidas, es utilizado por muchos científicos en la actualidad.

- Puede ser usado para preparar documentos por cualquier autor con un procesador de textos. Lo único necesario es utilizar un método de marcado formado por caracteres ASCII.

- Si no se desea crear una DTD por considerarse una difícil labor, el usuario puede utilizar una de las muchas comercializadas actualmente.

- El fichero producido soportará con flexibilidad una gran cantidad de formas de impresión y gestión electrónica y podrá ser usado para diferentes propósitos.

3.2 O.D.A.

ODA [11] [12] (**Office Document Architecture**) como SGML, es un estándar para la representación e intercambio de documentos estructurados.

ODA provee un modelo de documento jerárquico y orientado a objetos. Cada objeto se corresponde con un componente del documento, y los atributos contienen información acerca de los objetos. Un documento es considerado como un árbol, donde la estructura está definida por la forma del árbol y el contenido se almacena enteramente en sus hojas.

ODA asigna dos estructuras al texto dependientes entre sí y que coinciden a nivel de contenido: una estructura lógica que describe las relaciones abstractas entre componentes del texto y una estructura de formato definida en función de páginas, columnas y márgenes.

El estilo de un documento ODA después de impreso, viene determinado por un proceso que da forma al texto. Este proceso se encarga de dividir el contenido en bloques de texto y dependiendo de los atributos de cada objeto, coloca cada bloque en su lugar adecuado en una página.

4. Proceso de etiquetado automático.

4.1 Objetivos y definición del problema.

En las secciones anteriores se han presentado las ventajas y posibilidades ofrecidas por las nuevas tecnologías al proceso de publicar un documento. La principal conclusión de todo ello podría resumirse en que si la comunicación entre los participantes en el proceso de publicación y el tratamiento de la información se hiciera electrónicamente, todo el proceso se mejoraría.

Este trabajo trata de aplicar los resultados anteriores a la publicación automatizada de artículos científicos. Para su desarrollo se toman como base los siguientes puntos [1]:

- La estructura generalmente común de los artículos científicos: título, autor, abstract, cuerpo de texto y referencias.

- La posibilidad de cualquier autor actual de producir documentos en formato ASCII y enviarlos vía e-mail o discos flexibles.

- La actual aceptación de los editores de texto marcado en algún sistema que ellos suministran, y que pueden traducir directamente a código PostScript interpretable por las máquinas de composición.

- La expansión de los programas de autoedición y su aceptación como sistemas de marcado por los editores. En concreto, en este trabajo se adoptó el programa Xerox Ventura Publisher, el cual marca el texto insertando etiquetas delante de cada párrafo.

- La posibilidad de representar con un sistema el conocimiento necesario para reconocer cada una de las partes que constituye un artículo. En el futuro, se espera que los sistemas de publicación por ordenador contengan conocimiento en áreas tales como el estilo de formar una página, la clasificación de documentos y quizá, el diseño de estilos [13].

Con todo ello, el propósito de este trabajo es desarrollar un primer prototipo de sistema basado en el conocimiento capaz de aceptar artículos científicos sin forma definida y contenidos en ficheros ASCII, e insertar automáticamente, las etiquetas adecuadas para Ventura Publisher.

El programa siempre insertará las mismas etiquetas que indicarán la estructura lógica del documento. Las diferencias en el estilo final del texto dependerán de la hoja de estilo que esté definida en Ventura en ese momento. Así, el autor podrá editar su texto como si fuera el propio editor, si este último le suministra la hoja de estilo adecuada (cada editor tiene definida su propia hoja de estilo dependiendo de las características de su publicación).

Es necesario señalar que el prototipo que se quiere desarrollar no se trata de uno de los ya comercializados sistemas de reconocimiento óptico de caracteres (OCR) capaces de reconocer ademas de los caracteres de una página, su formato (fuente, estilo, etc.). No se puede asociar tampoco este sistema a los traductores de formato entre procesadores de texto, puesto que en nuestro caso el documento inicial no se encuentra más que en puro ASCII y sin más referencia a su forma o estilo que su propia estructura.

Cómo resultado se proporcionará un medio de preparación de artículos científicos (transmisión, edición y formato) tan útil para el editor como podría ser el recibirlo ya formateado en LaTeX o SGML pero sin los inconvenientes y restricciones que suponen estos métodos para el autor.

El prototipo reconocedor de texto ha sido implementado en PROLOG principalmente por dos razones:

- Por ser un lenguaje muy adecuado para la elaboración de prototipos, ya que permite al programador abordar el problema a diferentes niveles de abstracción según su conocimiento del mismo.

- Tratarse de un proceso de análisis sintáctico y semántico del texto que requiere una descripción del objeto del problema más que una solución particular.

Como complemento al principal objetivo ya mencionado se ha implementado un tratamiento de las referencias y citaciones del texto que las localiza y numera automáticamente.

4.2 Arquitectura del sistema y metodología de implementación.

El artículo en formato ASCII que va a ser formateado por el sistema debe sufrir varias transformaciones a lo largo de la arquitectura del mismo.

• Eliminación del formato ASCII y Separación de Párrafos.

Este proceso consiste en: eliminar los caracteres de retorno de final de línea y dejar los de final de párrafo y reducir cada secuencia de blancos a uno solo.

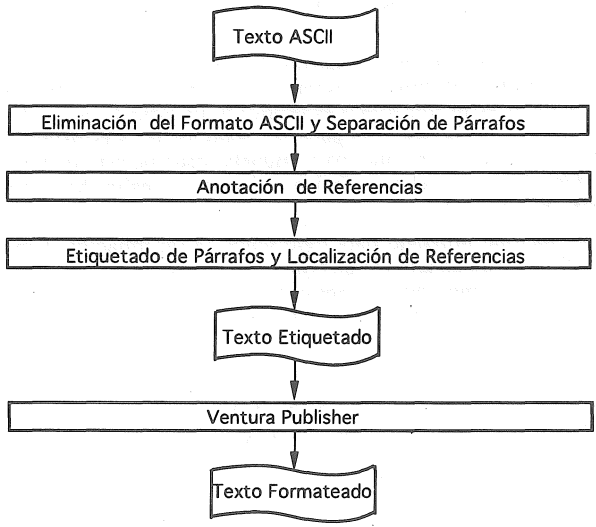
La implementación de esta primera transformación se ha hecho en lenguaje C ya que conlleva un simple análisis léxico. Cada carácter es analizado individualmente, sin tener en cuenta su posición en la estructura del documento. El único conocimiento que se necesita, es la posición del carácter en la línea y cuales son los que le siguen.

• Anotación de referencias.

Después de recorrer todo el texto hasta la sección de referencias, se hace una lista enumerada de todas las referencias que aparecen en la mima.

Esta segunda pasada sobre el artículo ha sido implementada en PROLOG y produce la inserción en la base de conocimiento de PROLOG de un hecho por cada una de las referencias interpretadas con el número que se le asocie.

La posterior conexión entre referencias y citaciones se hace por medio del nombre del autor y su año de publicación. Ambas informaciones han de ser extraidas de todos los datos que conforman una referencia.



- **Etiquetado de Párrafos y Localización de Referencias.**

Como se ha dicho anteriormente, Ventura etiqueta cada párrafo de texto. Por esta razón, definimos los párrafos como elementos básicos.

Este es el proceso clave del sistema y lo componen dos tareas que se ejecutan simultáneamente:

- Verificar la estructura el texto, reconociendo cada una de sus componentes.
- Substituir en el mismo cada una de las citaciones por el número asociado con la referencia correspondiente.

Para verificar la estructura del texto el programa debe localizar la posición del párrafo en la preestablecida estructura del artículo dependiendo del texto previamente recorrido. Su código consiste en una representación interna, por medio de reglas (PROLOG), del posible rango de elementos que pueden formar un documento; es decir, la implementación del programa se basa en el esquema que representa la estructura de un artículo científico, donde cada elemento es un párrafo diferente. En la siguiente sección, se habla de la representación de esta estructura utilizando una gramática libre de contexto y de su transformación en reglas.

El C utilizado para escribir la primera parte del prototipo fue Turbo C sobre un ordenador compatible PC. Posteriormente, este programa fue transmitido al mismo sistema en que fueron escritos los programas en PROLOG. Se trata de un VAX cluster con sistema operativo VMS. El PROLOG utilizado fue el incluido en el entorno POPLOG, desarrollado en la Universidad de Sussex en Inglaterra [14].

4.2.1 Representación gramatical de la estructura.

La especificación lógica del problema, por medio de una gramática libre de contexto, supuso la primera aproximación a la implementación del sistema en PROLOG.

La representación estructural del documento es una visión de alto nivel de su organización lógica. Esta representación puede considerarse a dos diferentes niveles: específico y genérico. La **estructura lógica genérica** representa el modelo para los documentos de una clase (p.ej.: cartas, artículos, manuales, etc). La **estructura lógica específica** representa una instancia en una clase de documento [15].

El sistema basado en el conocimiento desarrollado, es una representación gramatical de las relaciones entre los **objetos atómicos** (párrafos) que definen la estructura de un artículo científico. De esta forma, la estructura lógica específica de un artículo es el árbol que se produce al filtrar su texto. Este árbol forma la base para guiar la traducción de la estructura lógica del artículo a su apariencia física, o lo que es lo mismo, para etiquetar cada párrafo de texto. El razonamiento efectuado por el sistema basado en el conocimiento siempre se realiza hacia adelante, junto con el backtracking estándar del PROLOG.

Por dar una idea del método utilizado para representar la información, supongamos que en la posible estructura definida para un articulo científico, el abstract lo constituyen un conjunto de uno o más párrafos de texto comprendidos entre el título del artículo y su cuerpo principal. Opcionalmente, el abstract podría ir precedido de la palabra 'Abstract'. Una manera de reconocer un párrafo de texto como parte del abstract sería:

> **SI** (Párrafo después de la palabra 'ABSTRACT' **OR** Párrafo después de Título) **AND** Párrafo antes de Cuerpo de Texto
>
> **ENTONCES** Párrafo es Abstract.

4.3 El fichero de entrada.

El fichero de entrada, enviado por el autor, debe cumplir ciertos requerimientos [1]:

- Guardar el formato ASCII y sólo contener caracteres ASCII.
- Contener, al menos, dos retornos al final de cada párrafo.
- Tener la estructura especificada.

Para unos resultados óptimos, el autor debe tener en cuenta los siguientes puntos dependiendo de los resultados que quiera obtener:

l) El texto del fichero etiquetado y su orden, coincide exactamente con el de entrada.

2) Los nombres de todos los autores deben aparecer en un sólo párrafo y sus direcciones van en párrafos independientes y en el mismo orden que los nombres.

3) El abstract debe ir precedido siempre de su título (la palabra "ABSTRACT").

4) La numeración de los títulos de secciones y subsecciones debe llevar un orden decimal lógico. Los títulos sin numerar deben ser de menos de cincuenta caracteres y estar formados de una sola frase. Nunca terminarán en coma, punto y coma o dos puntos.

5) La última sección debe ser la de referencias empezando por su título (la palabra "REFERENCES").

6) Los símbolos que identifican un párrafo de numeración son: '*', '.', '-', '(', '[' (p.ej.: (1), [ii], etc.) y un número o letra seguidos de ')' o '-'. Todos ellos deben ir seguidos de un espacio en blanco. Un número seguido de un punto identifica un título y nunca un párrafo de numeración.

7) Todos los elementos del artículo deben guardar la estructura especificada previamente, hecho que como se puede observar no presenta ninguna restricción, ya que es la usualmente utilizada.

4.3.1 Gráficos, figuras y tablas.

El sistema desarrollado sólo trata con texto, el resto de elementos deben ser provistos por el autor en ficheros separados. Posteriormente, un operador de Ventura creará un marco e insertará en él el fichero con la figura (es el único procedimiento manual que queda en el estado actual del sistema).

La posición exacta del gráfico o figura, debe ser especificada por el autor en un párrafo especial de su texto. Sus dos primeros caracteres serán la secuencia '=>' seguida del nombre del fichero a insertar. El siguiente párrafo de texto es considerado un pie de figura si viene precedido por los símbolos '<='. En ambos casos, se inserta una etiqueta especial para señalar al operador la necesidad de su inserción.

Esta idea puede ser ampliada a textos en los que su forma se relaciona con su contenido (poemas, programas, etc).

4.4 Tratamiento de referencias.

Durante el desarrollo de la investigación, se consideró que el programa dispusiera de una facilidad opcional para manejar las referencias con las siguientes utilidades:

- Reconocimiento de las citas a referencias del texto.
- Inserción del número que el programa asocia con la referencia en el lugar adecuado y entre corchetes.
- Un etiquetado especial para las referencias de la sección correspondiente.

La aplicación trabaja con una sola clase de referencias cuya forma es:

1) Nombre de autor/es con una coma al final.

2) Nombre del libro u otro tipo de documento entre comillas.

3) Cualquier otra información incluyendo en cualquier posición el año de publicación siempre entre paréntesis.

En una referencia, el nombre de cada autor será cualquier combinación de nombres, iniciales y apellidos. Varios autores irán separados por comas y los dos últimos por la palabra 'and' o el símbolo '&'. El autor debe tener en cuenta que el último nombre propio o apellido completo que aparezca en un nombre de autor (no iniciales), será considerado el **nombre de citación**. A continuación se enumeran un conjunto de referencias cuyo nombre de citación podría ser [Smith (1988)]:

Smith, (1988) "Electronic Publishing". Ed. Benjamin. London.

J. Smith, "Electronic Publishing", (1988). Ed. Benjamin. London.

John Smith, "Electronic Publishing". Ed. Benjamin. London (1988).

John J. Smith, (1988) "Electronic Publishing" Ed. Benjamin. London.

Para identificar una cita a referencia en el texto, el autor debe especificar el nombre de citación y el año de publicación entre paréntesis y, ambos, entre corchetes. El programa reconocedor de párrafos, sustituye cada citación por el adecuado número de referencia, al mismo tiempo que recorre el texto. Previamente, el programa debe identificar y numerar todas las posibles referencias de la sección de referencias. Entre otras, es posible reconocer en el texto las siguientes citaciones:

[Smith (1988)]	[Smith et al (1990)]
[Smith & Hole (1989)]	[Smith, Hole and Jones (1991)]

En el caso de dos autores en una referencia, el nombre de citación está compuesto por los dos últimos nombres completos de cada autor con 'and' o '&' entre ellos. Para más de dos autores, el autor del artículo puede escribir en la sección de referencias o en las citaciones todos sus nombres o sólo el de uno de ellos y las palabras et al. En el primer caso, el autor puede opcionalmente poner

todos los nombres de citación o usar el primero de ellos y las palabras et al. En el segundo caso, la citación debe consistir del nombre de citación que aparezca y et al seguido de la fecha. De cualquier forma 'and' y '&' son intercambiables.

Si por algún error, una citación no se corresponde con ninguna referencia, el sistema no modifica el texto pero da un mensaje de error al usuario.

El etiquetado de la sección de referencias se diferencia de los demás en que las comillas que rodean al título son sustituidas por los códigos de Ventura que hacen aparecer el texto en cursiva.

La principal ventaja de utilizar la numeración automática de referencias, es que el autor puede insertar, borrar o modificar con la seguridad de que al final no habrá errores en la numeración. Las referencias, dentro de su sección, pueden estar ordenadas al gusto del autor. Su numeración es hecha por el sistema en el mismo orden en que aparecen.

5. Limitaciones del sistema.

El sistema desarrollado, entre otras cosas, trata de ser un trabajo en el que se muestren las grandes posibilidades que los métodos de inteligencia artificial ofrecen para solucionar problemas en el área presentada. En el limitado tiempo asignado para su desarrollo, se trató de ofrecer un mayor número de posibilidades, a cambio de restarle operatividad al sistema; como consecuencia, el sistema presenta algunas limitaciones [1]. Entre ellas:

- El ASCII que reconoce está reducido al conjunto básico de caracteres (ASCII del entorno POPLOG). Sin embargo, cualquier código ASCII entre los símbolos '< >' será traducido por Ventura en su correspondiente carácter.

- Dado que algunos procesadores de texto traducen los tabuladores a espacios, el sistema no los acepta ya que los reduce a un sólo espacio.

- Al igual que Ventura, el sistema no acepta notas a pie de página.

- Los errores en la estructura del texto o en la enumeración de títulos, figuras o párrafos no son tratados.

- Los métodos de numeración y los caracteres especiales para marcado de párrafos son limitados.

- El sistema debería haber sido implementado para su funcionamiento sobre un PC-compatible (la versión de Ventura Publisher que se utilizó era para PC). Sin embargo, por tratarse de un prototipo esto se ha sacrificado en virtud del mejor aprovechamiento de los recursos disponibles.

Como puede observarse, algunas de estas limitaciones pueden ser fácilmente eliminadas y otras necesitan una mayor investigación que podría ser tema de un futuro trabajo.

6. Conclusiones.

El prototipo resultante de esta investigación trabaja exitosamente para los artículos que se ajustan a la estructura especificada por el sistema y que cumplen ciertas reglas, actualmente empleadas y por tanto no restrictivas, en el proceso de mecanografiado en el ordenador.

El sistema en las últimas etapas de su desarrollo se ha aplicado a diversos textos ASCII en Middlesex University de Londres, para el formato de presentación de la revista científica *"The Computer Journal"* que allí se edita. Los resultados obtenidos coinciden con los maquetados manuales, ofreciendo el sistema además de rapidez y seguridad, la opción de tratamiento de referencias, que se ha mostrado muy útil.

En futuros trabajos se considerará el abordar muchas más estructuras de documentos, así como posibilitar más técnicas de etiquetado. El resultado será un formateador inteligente de texto que

permita editar documentos mecanografiados en diferentes procesadores de texto y, posteriormente, publicarlos en una amplia gama de formas definidas por el usuario, disponiendo para ello de varios lenguajes de marcado.

7. Agradecimientos.

Este trabajo ha sido llevado a cabo en Middlesex University de Londres gracias a la concesión por la Universidad Politécnica de Valencia de una beca ERASMUS.

Así pues, deseo agradecer a la Universidad Politécnica de Valencia y al proyecto europeo para el intercambio de estudiantes (ERASMUS), la oportunidad que nos dieron de desarrollar este trabajo, y a Middlesex University por su buena acogida y medios prestados.

También quiero dar las gracias a mis compañeros en Middlesex University y a mi director y colaborador Peter Hammersley, por su confianza y apoyo. Por último, doy gracias a Francisco Toledo de la Universitat Jaume I de Castellón por su ayuda y supervisión al redactar este artículo.

Referencias.

[1] Aramburu, M. J. (1991): *Tagging of Scientific Papers: An Expert System to Prepare ASCII Text for Publication by Ventura*: Middlesex Polytechnic, London.

[2] Hammersley, P. (1985): "The Use of Generic Coding in the Produccion of Journals and Conference Proceedings", *International SGML Conference*, Manchester.

[3] Goldfarb, C. F. (1990): *The SGML Handbook*: Ed. Claredon Press, Oxford.

[4] Donald, A y Knuth, E. (1984): *The TEXbook*: Ed. Addison-Wesley Publishing Company, USA.

[5] Ossanna, J. (1982): *Nroff & Troff. Users Manual and Addendum to the Troff Users Manual*: Bell Laboratory, USA.

[6] Holzgang. (1989): *PostScript Programmers Reference Guide*: Ed. Scott, Foresman and Company, USA.

[7] Reid, C. G. (1988): *PostScript Language. Program Design Adobe Systems Incorporated*: Ed. Addison- Wesley Publishing Company, USA.

[8] Lamport, L. (1988): *Document Production: Visual or Logical?*: Ed. TUGboat.

[9] Bryan, M. (1988): *SGML an Authors Guide to the Standard Ceneralized Markup Language*: Ed. Addison- Wesley Publishing Company, UK.

[10] Smith, J. M. (1986): "The Implications of SGML for the Preparation of Scientific Publications", *The Computer Journal*, Vol. 29, Nº 3: Ed. Cambridge University Press, UK, 193-200.

[11] Brown, H. (1989): "Standars for Structured Documents", *The Computer Journal*, Vol. 32, NQ 6: Ed. Cambridge University Press, UK, 505-514.

[12] Guardalben, G y Giacomello, M. (1990): "An ODA Page Planner for Professional Publishing EP90", *Proceedings of the International Conference on Electronic Publishing, Document Manipulation & Typography*: Ed. Cambridge University Press, USA.

[13] Kist, J. (1987): *Electronic Publishlng. Looking for a Blueprint*: Ed. Croom Helm, UK.

[14] Systems Designers (1986): *POPLOG User Guide*: Ed. University of Essex, UK.

[15] Furuta, R., Quint, V. y André, J. (1988): "Interactively Editing Structured Documents", *Electronic Publishing-odd*, Vol. 1(1): Ed. John Wiley & Sons, UK, 19-44.