Ingeniare. Revista Chilena de Ingeniería
ISSN: 0718-3291
facing@uta.cl
Universidad de Tarapacá
Chile

Henríquez Miranda, Carlos; Guzmán, Jaime
Extracción de información desde la web para identificar acciones de un modelo de dominio en planificación automática
Ingeniare. Revista Chilena de Ingeniería, vol. 23, núm. 3, 2015, pp. 439-448
Universidad de Tarapacá
Arica, Chile

Extracción de información desde la *web* para identificar acciones de un modelo de dominio en planificación automática

Information extraction from the web to identify actions of an automated planning domain model

Carlos Henríquez Miranda[1] Jaime Guzmán[2]

Recibido 22 de noviembre de 2013, aceptado 9 de diciembre de 2014
Received: November 22, 2013 Accepted: December 9, 2014

RESUMEN

La Planificación Automática (PA) es la disciplina de la Inteligencia Artificial que busca la producción de una secuencia de acciones que permiten alcanzar un objetivo específico, y que requiere la definición de un modelo de acción como flujo de entrada. Sin embargo, comenzar la construcción de un modelo de este tipo es una tarea difícil incluso para expertos. Este trabajo propone extraer información desde la web para luego identificar los elementos que corresponden a un modelo de acción para tareas de PA, lo que se busca es analizar un conjunto de páginas web donde se encuentra información en relación con planes ya producidos que resuelven un problema particular y extraer de allí un conjunto de pasos que permitan la solución a una problemática planteada. Después de recuperarlos se procesarán usando herramientas de Procesamiento de Lenguaje Natural (PLN), para así identificar un conjunto de acciones que hagan parte de un modelo de dominio en PA. El sistema propuesto alcanzó en promedio una precisión del 89,87% logrando, además, guardar todas las acciones extraídas en una ontología para formar una gran Base de Conocimiento (BC) que permite más tarde utilizarlas para otros dominios en PA. En este artículo se presenta el resultado de investigación parcial del uso de las herramientas de extracción, preprocesamiento, identificación de componentes y almacenamiento en la ontología.

Palabras clave: Extracción información, planificación automática, ontología, *web*, PLN, PDDL.

ABSTRACT

Automated Planning (AP) is the discipline of Artificial Intelligence whose aim is to produce a sequence of actions in order to achieve a specific goal and requires the inflow of a pre-defined action model. However, to start the construction of this model is a difficult task even for experts.
This paper proposes the extraction of information from the Web, in order to identify those elements that correspond to the action model for AP tasks, by analyzing a number of Web pages that contain data of already existing programs and extracting from them a set of tools which would allow to solve any given problem. The data obtained will be processed using Natural Language Processing (NLP) tools, in order to identify a set of actions that are part of an AP domain model. The proposed system reached an average accuracy of 89.87% and save all actions taken on ontology to form a Knowledge Base (BC), which allows later use for other domains of PA. This article presents the results of a partial research about the uses of extraction tools, pre-processing, identification of components and storage in the ontology.

Keywords: Information extraction, automated planning, ontology, Web, NLP, PDDL.

[1] Universidad Autónoma del Caribe. Barranquilla, Colombia. E-mail: chenriquez@uac.edu.co
[2] Universidad Nacional de Colombia. Medellín, Colombia. E-mail: jguzman@unal.edu.co

INTRODUCCIÓN

Del área Inteligencia Artificial (IA) se desprende el concepto de planificación, que según [1] es el proceso de búsqueda y articulación de una secuencia de acciones que permiten alcanzar un objetivo. La planificación busca producir planes para que sean usados por humanos o agentes inteligentes. Para lograr que esta producción se realice automáticamente (mediante un planificador) hay que definir varios elementos como: (i) estado inicial, que es la situación de partida, (ii) la meta que describe las condiciones que se tienen que dar para considerar por terminado el proceso, (iii) las acciones que transforman un estado a otro, (iv) el plan mismo, que es el conjunto de acciones que permite pasar del estado inicial al estado final y (v) las heurísticas, que es el conocimiento que permite obtener de forma eficiente un plan. Algunos elementos como las acciones, hacen parte del modelo de acción conocido como modelo de dominio. Otros elementos como el estado inicial hacen parte del problema a resolver por un planificador. Ambas especificaciones son necesarias para lograr el proceso de planificación, sin embargo, la construcción de estos modelos a partir de cero es una tarea difícil aun para un conjunto de expertos.

En este trabajo se propone identificar inicialmente un conjunto de acciones y otros elementos a partir de recursos encontrados en la *web*. En esta se encuentra una enorme cantidad de recursos representados en diferentes fuentes y formatos, gran parte de estos recursos se presentan en lenguaje natural en forma de *blogs, wikis* o redes sociales [2], otras en formas semiestructuradas como los que ofrecen ventas de productos o servicios. Por esa variedad de presentaciones y estructuras, recuperar información desde estos recursos se vuelve una tarea difícil y a veces imposible. El área conocida como Recuperación de Información (RI) aborda este problema encontrando documentos relevantes desde un gran repositorio en respuesta a un criterio definido [3]. Como existe mucha información, se han creado buscadores que se encargan de recuperarla por intermedio de consultas clave [4]. Pero no solo es traer documentos relevantes de una consulta específica, ya que en la *web* cualquier individuo puede brindar información valiosa en diferentes áreas como la economía, industria, medicina, robótica, entre otras, sino que debe existir otro tipo de sistemas apoyados en técnicas de inteligencia artificial (IA) que se encarguen de buscar dentro de los documentos, explorar su contenido y extraer información pertinente de un tema en particular. Estas nuevas herramientas se enmarcan en el área conocida como Extracción de información (EI) que se ocupa de estructurar los contenidos dentro de los textos que son relevantes para el estudio de un dominio particular [5]. En otras palabras, el objetivo de un sistema de EI es encontrar y enlazar la información relevante, mientras ignora la extraña e irrelevante [6].

Más concretamente una tarea de EI es definida por el documento de entrada y el objetivo de extracción. La entrada pueden ser documentos sin estructura como texto libre, como en la Figura 1, escritas en lenguaje natural o documentos semiestructurados que se presentan como tablas o listas detalladas y enumeradas, como en la Figura 2, y lo que se extrae de allí puede ser nombres de personas, países, años, autos, ocupaciones, entre otros; siempre dependiendo del criterio específico que represente las necesidades de búsqueda.

The causes of cancer are diverse, complex, and only partially u
including tobacco use, dietary factors, certain infections, expos
pollutants.[2] These factors can directly damage genes or comb
mutations.[3] Approximately 5–10% of cancers can be traced di
by not smoking, eating more vegetables, fruits and whole grains
weight, exercising, minimizing sunlight exposure, and being vac

Cancer can be detected in a number of ways, including the pres
imaging. Once a possible cancer is detected it is diagnosed by
with chemotherapy, radiation therapy and surgery. The chances
cancer and the extent of disease at the start of treatment. Whil
more common in children, the risk of developing cancer general
deaths worldwide (7.9 million). Rates are rising as more people
world.[6]

Figura 1. Información en lenguaje natural.

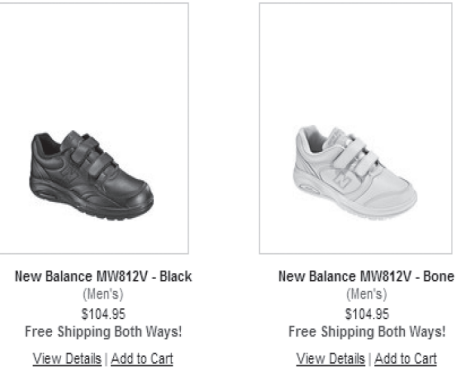

Figura 2. Información semiestructurada.

En los trabajos más representativos de la literatura se han abordado diferentes dominios de información, usando múltiples formas para llevar a cabo el proceso de EI. Por ejemplo, en [7] se describe un enfoque para análisis de la comunicación empresarial específicamente extrayendo información de los correos electrónicos, por su parte en [8] se ubican entidades y sus respectivos conceptos a partir de la exploración de las tablas en un documento HTML. En [9] se presenta un paradigma de extracción que facilita el descubrimiento de relaciones extraídas de texto, independiente del dominio y del tamaño del recurso *web*. En [10] se explota la apariencia visual de la información impulsado por relaciones espaciales que se producen entre los elementos de una página HTML. En [11] se construyen herramientas robustas para la extracción de información *web* ante cambios en la estructura de la página basadas en un modelo de costo mínimo y en [12] se propone un enfoque de descubrimiento de patrones para la rápida generación de extractores de información. Finalmente en [13] y [14] se muestra cómo desde un sitio *web* se obtienen un conjunto de acciones referentes a planes que luego se convierten en elementos de dominio de PA específicamente para el lenguaje de definición PDDL.

PDDL es un lenguaje centrado en la acción, inspirado en el modelo STRIPS para formular problemas de planificación. Se ha convertido en el lenguaje estándar desde 1998 utilizado en la competición internacional de planificación (ICAPS). Para especificar una tarea en PDDL se identifican dos archivos, el primero define el dominio que describe los predicados y las posibles acciones, el segundo detalla el problema donde se describen los objetos, el punto inicial y la meta. [15]

A partir de [13] y [14] se realizó una primera aproximación en [16] y específicamente en este trabajo usamos otras técnicas de PLN y almacenamos lo extraído en una ontología. El uso de ontologías como BC mejora lo propuesto en [13], [14] y [16], ya que se consigue un gran avance en los procesos de adquisición y gestión del conocimiento asociado a los dominios de planificación, problemática que es tratada por [17]: *la comunidad de planificación debe estudiar las técnicas y herramientas relacionadas con la gestión de conocimiento desarrolladas en otras áreas, y estudiar cómo adaptarlas e integrarlas en los sistemas de planificación*, además se pueden generar nuevos planes por recombinación aprovechando el conocimiento almacenado, así como también poseer un vocabulario expresivo con el que se puedan definir nuevos problemas de planificación planteando nuevos estados y acciones [18]. Almacenar las acciones en una ontología permite contar con una potente herramienta para realizar interesantes inferencias durante el proceso de búsqueda de planes, lograr mayor expresividad para modelar dominios complejos y usar planificadores potentes previa traducción a un lenguaje de planificación específico.

Para las tareas de extracción se usa un *wrapper*, programa especializado que identifica los datos de interés sobre la base de varias reglas gramaticales, que luego de extraerlos los transforma a una estructura de datos apropiada para su posterior manipulación [19]. Particularmente el proceso consiste en una exploración al documento *web* escogido y el *wrapper* revisa toda su estructura HTML en busca de información relevante dependiendo de patrones inicialmente definidos. Con el *wrapper* se combinan servicios *web* basados en RestFul que permiten tener diseños más simples, bajo consumo de recursos, *URI* por recurso y generalmente por ser servicios fáciles de construir y adoptar [20-22].

Adicionalmente se exploran herramientas para PLN para tareas de *stopwords, stemming* y análisis morfológico [23-24]. El resto del artículo se organiza de la siguiente manera: en la próxima sección se propone un modelo para identificar las acciones, en la siguiente parte del texto se muestran los resultados de un prototipo y finalmente se presentan las conclusiones.

IDENTIFICACIÓN DE ACCIONES

Para el proceso de extracción se toma como referencia el recurso *web* conocido como *WikiHow*[3], un repositorio que contiene manuales con más de 150 mil artículos, donde se encuentran textos escritos por un grupo de colaboradores que explican paso a paso cómo llevar a cabo ciertas tareas. Una página de *WikiHow* consta de título, descripción, categoría, ingredientes y un conjunto de pasos (ver Figura 3). Para el proceso de extracción se usa la Tabla 1, que describe de dónde se va a extraer cada componente.

[3] www.wikihow.com

Figura 3. Recurso *WikiHow*.

Tabla 1. Comparación entre la *web* y planificación.

Recurso *web*	**Planificación**
Page	Plan
Title	Objetivo final
Ingredients	Objetos
Steps	Acciones
Category	Categoría

Cada página analizada representa un posible plan compuesto por una secuencia de pasos que permiten alcanzar un objetivo deseado. Cada plan pertenece a una categoría del recurso *WikiHow*.

Para todo el proceso se propone un modelo inicial presentado en la Figura 4, dividido en cinco componentes: el primero (*Ext-Web Servicie*), que recibe una URL (dirección donde está la página) y junto con otros componentes entrega una trama de entidades al componente *Pre-Processing*. El segundo (*Browse-Cat*), toma una página (formato HTML) que contiene una categoría de la *WikiHow* y obtiene todos los planes de esa categoría. El tercero (*Wrapper-Plan*), recibe una página que contiene un plan y retorna la lista de pasos. El cuarto (*Pre-Processing*), toma la lista de pasos y realiza actividades correspondientes a un análisis léxico y, el último (*Build-plan*), toma las acciones y elementos identificados para poblar la ontología.

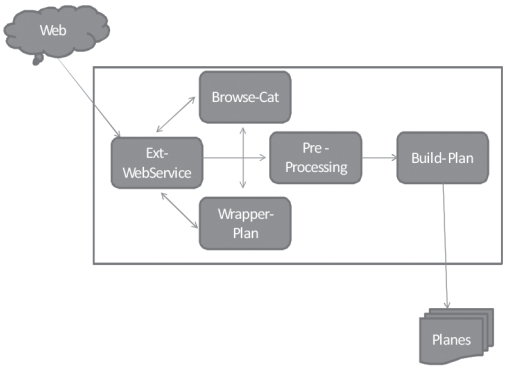

Figura 4. Modelo para identificar acciones.

El componente *Ext-Webservice* es la interface del modelo que ofrece dos servicios *web* [20]. El primero recibe un recurso que representa lo que se va a extraer y devuelve, dependiendo de la petición, una lista de pasos de un plan o una lista de artículos asociados a una categoría. Para hacer esto se apoya en los dos procesos subsecuentes *Wrapper-Plan* y *Browse-Cat*. El proceso puede hacerse tomando una categoría y procesando todos los planes de esa categoría o simplemente la extracción de los pasos de un plan. El componente *Browse-Cat* recibe como entrada una categoría, representada en una página *web* que contiene una lista de planes (ver Figura 5). A partir de esta página que es ingresada se obtiene

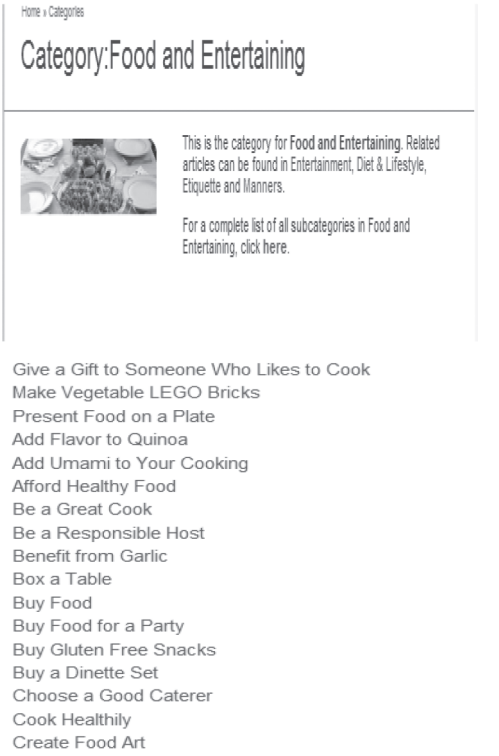

Figura 5. Categorías de *WikiHow*.

un conjunto de enlaces a páginas que representan los planes escritos en forma de pasos.

El *Wrapper-Plan* recibe la página a procesar de su predecesor y obtiene a partir de ella las entidades representadas en título, categoría, ingredientes y pasos. Para este componente se hace un análisis de la estructura de la página y se implementa un *wrapper* basado en la Tabla 1.

En la Figura 6 se muestra un ejemplo de un recurso de *WikiHow* (*Make-Bear-Paw-Cookies*) de donde se obtiene la información. En la Figura 7 se puede observar el resultado de la ejecución del *wrapper* cuando se le pasa la url: *http://www.wikihow.com/Make-Bear-Paw-Cookies*.

Figura 6. Pagina *WikiHow*.

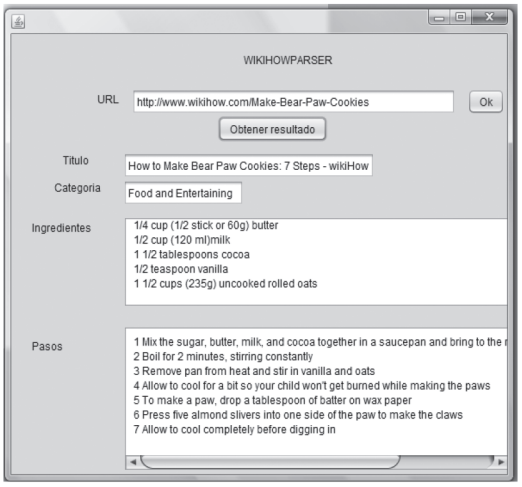

Figura 7. Extracción del *wrapper*.

Para el componente *Pre-Processing* se toma toda la traza enviada por el *Ext-Web Servicie* después del proceso que se hace en *Wrapper Plan* y realiza actividades correspondientes a un análisis léxico. Dentro de estas actividades se realiza primero un análisis morfológico en los pasos extraídos para clasificar cada entidad en alguna de las siguientes categorías: verbos (*VERB*), sustantivos (*NOUN*), adjetivos (*ADJECTIVE*) y adverbios (*ADVERB*). Para la implementación de este componente del modelo se usa la herramienta *Stanford POS Tagger* creada por investigadores del grupo de procesamiento del lenguaje natural en la Universidad de Stanford [25]. Luego de esta etapa, se limpian los datos borrando algunas palabras sin significado como artículos, pronombres, preposiciones, etc. Estas palabras en el idioma inglés se conocen como *stopwords* [26]. Adicionalmente para algunas entidades se utilizan herramientas de *stemming* para reducir una palabra a su raíz o lema.

El último componente *Build-Plan* toma las entidades restantes y las almacena en una ontología. En la Figura 8 se muestra una ontología básica que permite almacenar las entidades como plan, categoría, objetos y acciones.

Para lograr el proceso de almacenamiento, cada entidad tiene una correspondiente clase en la ontología: *Plan, Categoría, Objeto y Acción Extraída*. Estas son usadas para crear los respectivos individuos que permiten hacer el poblamiento de la ontología. Además, la ontología tiene las propiedades de cada clase, así como la relación entre cada una de ellas.

Con la información ya almacenada se podrán crear modelos de acción para un dominio de planificación en lenguaje PDDL (ver Figura 9) de la mano de un

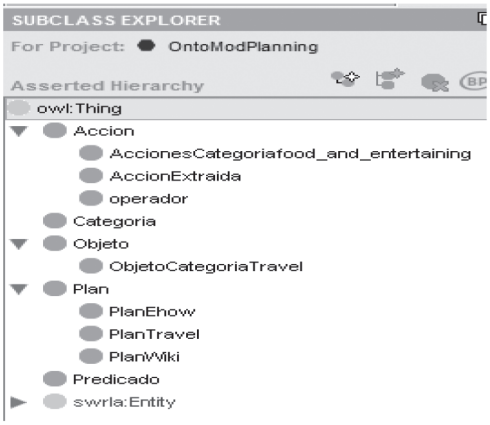

Figura 8. Extracción del *wrapper*.

```
(define (domain logistics)
(:types city place physobj - object
   package vehicle - physobj
   airplane truck - vehicle
   airport location - place
  )
(:predicates     (in-city ?loc - place ?city - city)
          (at ?obj - physobj ?loc - place)
          (in ?pkg - package ?veh - vehicle))

(:action LOAD-TRUCK
   :parameters     (?pkg - package ?truck - truck ?loc - place)
   :precondition   (and (at ?truck ?loc) (at ?pkg ?loc))
   :effect         (and (not (at ?pkg ?loc)) (in ?pkg ?truck)))

(:action LOAD-AIRPLANE
  :parameters (?pkg - package ?airplane - airplane ?loc - place)
  :precondition (and (at ?pkg ?loc) (at ?airplane ?loc))
  :effect       (and (not (at ?pkg ?loc)) (in ?pkg ?airplane)))
```

Figura 9. Parte de archivo PDDL dominio de logística.

experto. También se podrán realizar aplicaciones inteligentes que puedan llevar el conocimiento de la ontología como la representación de modelo de dominio directamente a un planificador.

EXPERIMENTO Y RESULTADOS

Para validar parte del modelo se hizo un prototipo que utilizó diferentes herramientas de programación. Este prototipo se hizo en Java 7.0 con uso de librerías especiales HttpClient 3.1 usando Netbeans IDE 7.11 bajo la arquitectura Rest usando *web services* y como servidor GlassFish 3.1.2. Para el proceso de extracción desde recursos *web* se usaron las bibliotecas de Jericho HTML Parser. Para la clasificación de entidades se usó la herramienta Stanford POS Tagger y tareas de *stemming snowball* [27]. Para el diseño de la ontología se utilizó protege 3.48 y para la manipulación, consultas y poblamiento las librerías de Apache Jena 2.10. Para el experimento se tomó *WikiHow* y se pasaron al modelo las direcciones (URL) de diferentes categorías (5) que organizan un conjunto de planes.

```
Computers and Electronics
 1 http://www.wikihow.com/Be-a-Technic
 2 http://www.wikihow.com/Buy-Used-Electronics
 3 http://www.wikihow.com/Clean-a-Touch-Screen
 4 http://www.wikihow.com/Do-Advanced-Computer-Yoga
 5 http://www.wikihow.com/Do-Computer-Meditation
 6 http://www.wikihow.com/Do-Computer-Yoga
 7 http://www.wikihow.com/Extend-a-Cheap-Modem%27s-Life
 8 http://www.wikihow.com/Keep-Your-Digital-Memories-Safe
 9 http://www.wikihow.com/Accept-That-Your-Computer-Is-Slow
```

Figura 10. Resultado categoría “Computers and electronics”.

En la Figura 10 se muestra el resultado específico de la categoría *Computers and electronics*, el que arrojó 201 planes.

Por cada categoría pasada se tomaron las URL de los planes, se exploraron y se extrajeron un grupo de pasos. En la Figura 11 se muestra el recurso *web How to Prepare Your Boat for Transport* al que se aplicó el proceso de extracción.

Para realizar el proceso de extracción de información asociada a lo que se requiere, se identificaron en cada recurso patrones en donde se localizaban el plan y sus pasos. Se revisaron muchas páginas similares y se obtuvo el patrón para la extracción. Por ejemplo para sacar el título, en la página se encuentra que este aparece rodeado de los *tags* <TITLE></TITLE>. Para los pasos se empleó un procedimiento similar. En la Figura 12 se muestra el resultado del proceso que obtiene el título y los pasos. Note que solo se extrae lo útil e importante según el fin específico.

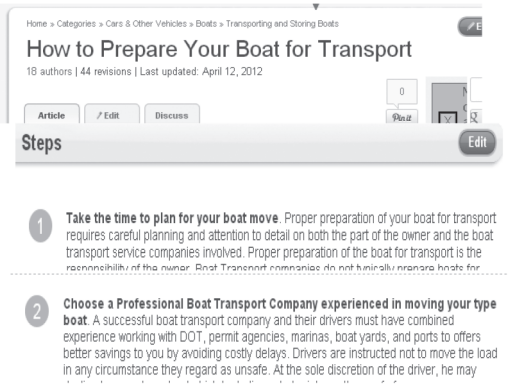

Figura 11. Página *WikiHow*.

```
Plan es  How to Prepare Your Boat for Transport: 20 steps
1 Take the time to plan for your boat move
2 Choose a Professional Boat Transport Company experienced in moving y
3 Communicate with your Boat Transportation Company
4 Measuring Your Boat for Transport
5 Understand how payment is to be made for both the boat transport ser
6 Remove and properly store the following items
7 Items to be removed inside your boat
8 Items to be removed outside your boat
9 Check for Zebra Mussels
10 Decide whether to shrink-wrap your boat or not
```

Figura 12. Extracción de pasos.

Con todo lo extraído se identifican de la traza las entidades de cada paso. Por ejemplo del paso 1, *Take the time to plan for your boat move* se dividen en entidades individuales así: *<<Take>> <<the >><<time>><<to>><<plan>><<for>><< your>><<boat>><<move>>*. Estas entidades son clasificadas en una categoría ya definida con anterioridad. Por ejemplo se obtienen las posibles **acciones** del **plan** como el **verbo** presente en la traza de cada paso (ver Tabla 1).

```
Entidades clasificadas por paso
Loading default properties from tagger models/wsj-0-18-left3words.tagge
Reading POS tagger model from models/wsj-0-18-left3words.tagger ... dor
Paso 1
Take VERB  the N/A  time NOUN  to N/A  plan NOUN  for N/A  your N/A  t
Paso 2
Choose VERB  a N/A  Professional NOUN  Boat NOUN  Transport NOUN  Comp
Paso 3
Communicate VERB  with N/A  your N/A  Boat NOUN  Transportation NOUN
Paso 4
Measuring VERB  Your N/A  Boat NOUN  for N/A  Transport NOUN
Paso 5
Understand NOUN  how ADVERB  payment NOUN  is VERB  to N/A  be VERB  m
```

Figura 13. Clasificación entidades.

En la Figura 13 se muestra el resultado de la clasificación morfológica de cada entidad. Las que no tienen ningún significado para el estudio aparecen como *N/A*. Después del proceso de clasificación encontramos que existen varios verbos identificados en algunos pasos, lo que acarrea problemas para la futura representación en un lenguaje de planificación (PDDL). Así que en el siguiente paso se eliminan las palabras que causan ruido (*StopWords*) y las que aparecen con *N/A* para que quede una traza de entidades más limpia.

Para la prueba de las herramientas creadas se examinaron un conjunto de categorías de *WikiHow* y los resultados se muestran en la Tabla 2.

Tabla 2. Resultados de extracción por categoría.

Categoría	Planes hallados	Pasos por plan
Pasatiempos	201	1803
Vida familiar	201	1681
En el trabajo	54	465
Arte y entretenimiento	201	1622
Viajes	93	892

Luego de todo el proceso de extracción se finaliza guardando las entidades restantes en la ontología como se muestra en la Figura 14. Esto se hace creando por cada entidad los individuos correspondientes a las clases en la ontología. Para determinar, por ejemplo, cuál es la acción se toma por cada paso la entidad que sea clasificada como verbo en la frase, esto se hace con la ayuda de su categoría gramatical. También para esta identificación se utiliza la clase operador de la ontología que contiene una infinidad de lista de verbos en inglés.

Adicionalmente se tomaron varios recursos de planes al azar, cuyo resultado se muestra en la Tabla 3.

A partir de los resultados obtenidos podemos decir que todas las categorías analizadas en el experimento arrojaron el conjunto de planes y acciones esperadas según el recurso brindado. El máximo de planes

Tabla 3. Resultados de extracción acciones por plan.

Plan	Entidades procesadas	Entidades borradas	% Eliminación	Acciones extraídas	Precisión
1	93	42	45,16	9	77,78
2	174	72	41,37	9	88,89
3	85	34	40	11	81,82
4	77	33	42,85	12	91,67
5	58	15	25,86	12	91,67
6	104	37	35,57	15	86,67
7	113	36	31,85	16	93,75
8	356	131	36,79	32	96,88
9	287	99	34,49	38	94,74
10	203	63	31,03	39	94,87

Figura 14. Clasificación de entidades.

hallados fue 201, que es el máximo número de planes por categoría y los números de pasos más altos encontrados fueron de 1.803. Para todos los planes hubo una extracción de la lista de pasos sin ningún inconveniente en todas las páginas procesadas. Al hacer un análisis *ad hoc* de forma manual de algunos planes de salida, se muestra que el sistema arrojó una precisión entre 77,78 % y 96,88% en acciones extraídas, lo que permite ver un buen comportamiento del sistema.

Después de la extracción se nota que las herramientas de clasificación y limpieza funcionan adecuadamente y que el porcentaje de entidades sin significado en los planes es alta (mayor que 26%). Es posible anotar que este es un trabajo inicial antes de pasar a las siguientes etapas necesarias que son el análisis sintáctico y análisis semántico específicamente para la desambiguación del sentido de las palabras. (*Build-Plan*).

Para finalizar consideramos que se ha logrado el objetivo en la extracción de acciones, la identificación de entidades y el poblamiento automático de la ontología. Sin embargo, cabe anotar que se encontraron ciertos problemas con la parte de la identificación de las entidades en 100%, como estaba previsto, y esto se debe a las mismas características que posee el lenguaje natural, de todas formas quedan por implementar etapas de análisis sintáctico y semántico que complementarían y mejorarían el proceso propuesto. De todas formas, las tecnologías que hacen parte de las herramientas usadas están sujetas a mejoras en el futuro. En cuanto al *wrapper* implementados cumplieron sus objetivos pero no están preparados para posibles cambios en la estructura HTML. Para eso se explorará el concepto de generación de *wrapper* automáticamente con técnicas inductivas (*wrapper induction*).

CONCLUSIONES

En este artículo se muestran los resultados iniciales de las herramientas utilizadas para cada uno de los procesos de un modelo propuesto para la extracción de información y la identificación propia de entidades útiles. Estas pretenden unirse al grupo de herramientas que permitirán la construcción de forma automática de planes a partir de la existencia del conocimiento disponible en la *web*. En los resultados obtenidos se puede observar que el proceso de extracción y análisis de información se hace de manera adecuada alcanzando en promedio un 89,87% de precisión en la extracción a partir del modelo propuesto. No obstante se presentan problemas en la identificación de la totalidad de las entidades.

Sin embargo, de todo lo extraído se logró almacenar las entidades identificadas (acciones) en una ontología, permitiendo contar con una potente herramienta para realizar interesantes inferencias durante el proceso de búsqueda de planes, lograr mayor expresividad para modelar dominios complejos y usar planificadores potentes previa traducción a un lenguaje de planificación específico.

En este trabajo también se puede percibir la importancia actual del área de la extracción de información en cuanto a la gran cantidad de conocimiento encontrado en la *web*. Esto debería ser aprovechado mucho más para recuperar y centralizar conocimiento valioso en áreas donde se participa activamente bajo un paradigma de colaboración entre expertos de cada línea de conocimiento.

Para trabajos futuros se propone mejorar el proceso de extracción de información que se hace con *wrapper,* haciendo uso de técnicas inductivas que permitan la recuperación de entidades independientemente del formato del recurso.

REFERENCIAS

[1] S. Russell y P. Norvig. “Inteligencia artificial un enfoque moderno”. Prentice Hall. Segunda edición. Madrid, España. 2004.
L. Pérez. “Redes Sociales, Blogs y Wikis: Tendencias y realidades”. 2010. Fecha de consulta: 7 septiembre de 2012. URL: www.slideshare.net/gentedeinternet/blogs-redes-sociales-y-wikis

[2] F. Martínez. "Recuperación de información: Modelos, sistemas y evaluación". EL KIOSKO JMC. 2004.

[3] O. Lobo y M. Dolores. "Métodos y técnicas para la indización y la recuperación de los recursos de la Worl Wide Web". Boletín de la Asociación Andaluza de Bibliotecarios. N° 57. 1999.

[4] A. Téllez. "Extracción de Información con Algoritmos de Clasificación". Tesis para optar al grado maestro de ciencias. Instituto nacional de astrofísica, óptica y electrónica. Tonantzintla, Puebla, México. 2005

[5] J. Cowie and W.Lehnert. "Information Extraction". Magazine Communications of the ACM .Vol. 39, Issue 1, pp, 80-91. January, 1996.

[6] M. Laclavík, S. Dlugolinský, M. Seleng, M. Kvassay, E. Gatial, Z. Balogh and L. Hluchý. "Email Analysis and Information Extraction For Enterprise Benefit". Computing and Informatics. Vol. 30, pp. 57-87. 2011.

[7] B. Dalvi, W. Cohen and J.Callan. "WebSets: Extracting Sets of Entities from the Web Using Unsupervised Information Extraction" WSDM '12 Proceedings of the fifth ACM international conference on Web search and data mining, pp. 243-252. 2012.

[8] O. Etzioni, M. Banko, S. Soderland and D. Weld. "Open Information Extraction from the Web". Magazine Communications of the ACM - Surviving the data deluge. Vol. 51 Issue 12, pp. 68-74. December, 2008.

[9] G. Penna, D. Magazzeni and S. Orefice. "Visualextraction of information from webpages". Journal of Visual Languages & Computing. Vol. 21, Issue 1, pp. 23-32. February, 2010.

[10] D. Liu, X. Wang, L.Li and Z. Yan. "Robust Web Extraction Based on Minimum Cost Script Edit Model". Web Information Systems and Mining Lecture Notes in Computer Science. Vol. 7529, pp. 497-509. 2012.

[11] C.H. Chang, C.N Hsu and S.C. Lui. "Automatic information extraction from semi-structured Web pages by pattern discovery". Journal Decision Support Systems-Web retrieval and mining. Vol. 35, Issue 1, pp. 129-147. April 1, 2003.

[12] A. Addis, G. Armano and D. Borrajo. "Recovering Plans from the Web". Proceedings of SPARK, Scheduling and Planning Applications woRKshop, ICAPS'09. 2009.
A. Addis and D. Borrajo. "From Unstructured Web Knowledge to Plan Descriptions". Information Retrieval and Mining in Distributed Environments Studies in Computational Intelligence. Vol. 324, pp. 41-59. 2011.

[13] M. Fox and D. Long. "pddl2.1: An Extension to pddl for Expressing Temporal Planning Domains". Journal of Artificial Intelligence Research. Vol. 20, pp. 61-124. 2003.

[14] C. Henríquez y J Guzmán. "Modelo de extracción de información desde recursos web para aplicaciones de la planificación automática". Prospectiva. Vol. 10 Nº 2, pp. 74-80. Julio - Diciembre 2012.

[15] S. Biundo, R. Aylett, M. Beetz, D. Borrajo, A. Cesta, T. Grant, T. McCluskey, A. Milani and G. Verfaille. "Technological Roadmap on AI Planning and Scheduling". Informe Técnico IST-2000-29656, PLANET, the European Network of Excellence in AI Planning. 2003.

[16] A. Ruiz. "Una aproximación ontológica al modelado de conocimiento en los dominios de planificación". Tesis para optar al grado de Doctor. Universidad Complutense de Madrid. España. 2010.

[17] A. Fernández . "Extracción de Información de la Web Basado en Ontologías". Tesis para optar al grado de Magíster. Instituto de Computación. Facultad de Ingeniería Universidad de la República. Montevideo, Uruguay. 2004.

[18] S. Tyagi. "Oracle. RESTful Web Services". 2006. Date of visit: June 6, 2012. URL: http://www.oracle.com/technetwork/ articles/ javase/index-137171.html

[19] R. Navarro. "Rest vs Web Service", pp. 5-10. 2006. Date of visit: June 15, 2013. URL: http://users.dsic.upv.es/~rnavarro/ NewWeb/ docs/RestVsWebServices.pdf

[20] A. Rodriguez. "RESTful Web services: The basics". 2008. Date of visit: June 10, 2013. URL: http://www.ibm.com/ developerworks/ webservices/library/ws-restful/

[21] L. Alonso. "Herramientas libres para PLN". 2005. Fecha de consulta: 10 de agosto de

2013. URL: http://www.cs.famaf.unc.edu.ar/~laura/freeNLP

[22] E. Méndez y J. Moreiro. "Lenguaje natural e indización automatizada". Ciencias de la Información. Vol. 30 Nº 3, pp. 11-24. 1999.

[23] K. Toutanova and C. Manning. "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora", pp. 63-70. 2000.

[24] R. Baeza-Yates and B. Rebiero-Neto. "Modern Information Retrieval". Addison Wesley, London, England. Chapter 1, pp. 9-15. 2003

[25] M. Porter. "Snowball: A language for stemming algorithms" 2012. Date of visit: December 11, 2012. URL: http://snowball.tartarus.org/texts/introduction.html