

CONSTRUCCIÓN DE PROTOTIPOS PARA LAS ETAPAS DE UN TRADUCTOR DE IDIOMAS BASADA EN EL ANÁLISIS Y ESTRUCTURACIÓN DE UNA PLATAFORMA TEÓRICA

RESUMEN

Este trabajo está enmarcado en el área del Procesamiento del Lenguaje Natural, específicamente la Traducción Automática. Se enumeran cada una de las etapas que hacen parte del proceso de traducción automática, documentando la base teórica y los resultados de prototipos que muestran el procesamiento realizado por cada etapa, permitiendo al lector diferenciar por medio de ejemplos los diferentes procesos que aquí intervienen. Concluye mostrando la salida del prototipo principal para frases afirmativas en presente simple del idioma inglés, teniendo como destino el idioma español, mostrando como se genera una salida libre de ambigüedades gracias al manejo de las relaciones temáticas.

PALABRAS CLAVES: Procesamiento del lenguaje natural, traducción automática, preprocesamiento, tokenización, análisis morfológico, diccionario, análisis sintáctico, análisis semántico.

ABSTRACT

The approach of this work is about the Natural Language Processing, specially the Machine Translation. It describes each of the stages of the Machine Translation process, documenting its theoretical base and the results of prototypes which demonstrate the processing made for each stage, letting the reader differentiate each process through examples. At the end it shows the output of the main prototype for affirmative sentences in present simple from English to Spanish, showing the output free of ambiguities thanks to the use of correlated topics.

KEYWORDS: *Natural language processing, machine translation, preprocessing, tokenization, morphological analysis, dictionary, syntactic analysis, semantic analysis.*

1. INTRODUCCIÓN

La traducción es un elemento fundamental para la comunicación de los seres humanos, más aún presentándose fenómenos como el de la globalización que obliga a un permanente intercambio ideas en diferentes lenguajes. El principal problema de los sistemas de traducción automática son las ambigüedades que se generan cuando se pasa al lenguaje destino. En la actualidad existen sistemas de traducción automática que permiten generar traducciones de muchos idiomas fuente a muchos idiomas destino y aunque la calidad de las traducciones de algunos son lo suficientemente buenas como para considerarse que pueden ser utilizadas de una forma seria, ha sido imposible evitar que se tengan salidas ininteligibles, pero que para la computadora es difícil determinar si son o no correctas.

En [2] y [6] se realiza una introducción a la traducción automática. En [7] se puede encontrar un compendio de gran parte de los diferentes sistemas de traducción actuales. En [8] se realiza una descripción detallada de la traducción automática. En [9] se hace un análisis del estado del arte de la traducción automática. En [10] se

puede encontrar una descripción del funcionamiento de los sistemas de traducción automática.

En este artículo se desarrolla una metodología para un prototipo de traductor que propone soluciones para cada una de las etapas que conforman el proceso de traducción automática, las cuales son preprocesamiento, tokenización, análisis morfológico, diccionario, análisis sintáctico, análisis semántico y síntesis al lenguaje destino. Para cada una de ellas, se documentan los problemas asociados y se describen los pasos seguidos para su solución, mostrando finalmente la manera como se obtuvieron traducciones no ambiguas mediante el uso de un motor de inferencia que hace uso de relaciones temáticas en la fase de análisis semántico.

2. PREPROCESAMIENTO

El preprocesador está compuesto por dos módulos, un filtro y un separador de contracciones.

2.1. Filtro

Mediante el empleo de expresiones regulares, el módulo de filtro es el encargado de compactar los separadores redundantes que existen en el texto.

JOSÉ FERNEY FRANCO B.

Ingeniero de Sistemas y Computación
Profesor Catedrático
Universidad Tecnológica de Pereira
ferney@utp.edu.co

CÉSAR FREDY GIL MEJÍA

Ingeniero de Sistemas y Computación
Ingeniero de Software
Microtec de Colombia Ltda.
fgil@utp.edu.co

SEBASTIÁN MARÍN LOAIZA

Ingeniero de Sistemas y Computación
Universidad Tecnológica de Pereira
petry@utp.edu.co

2.1.1. Modelamiento del módulo de filtro

Se definen los siguientes patrones de expresiones regulares:

Múltiples Enters: $(\backslash n \backslash s^*)$. El cual busca espacios redundantes entre saltos de línea. Se hace necesario que por lo menos haya un salto de línea seguido de cero o más espacios $(\backslash s)$.

Múltiples Espacios: $[\backslash t]+$. El cual busca espacios diferentes a Enter, esto para eliminar espacios redundantes entre palabras.

Espacios Iniciales: $(^\wedge[\backslash t]+)$. El cual busca uno o más espacios (tabulaciones o espacios en blanco) seguidos en el comienzo de una línea.

Separación Guión: $(-\backslash n)$. El cual busca un guión seguido de un salto de línea.

Para realizar el proceso de filtrado se realiza una substitución de las cadenas que cumplan con los patrones anteriores de la siguiente manera:

Múltiples Enters: $(\backslash n \backslash s^*) \rightarrow \backslash n$. Se substituyen por un solo salto de línea.

Múltiples Espacios: $[\backslash t]+ \rightarrow ' '$. Se substituyen por un solo espacio.

Espacios Iniciales: $(^\wedge[\backslash t]+)$ $\rightarrow ''$. Se substituyen por una cadena vacía.

Separación Guión: $(-\backslash n) \rightarrow ''$. Se substituyen por una cadena vacía, uniendo de esta manera en una sola palabra dos fragmentos separados por guión en un salto de línea.

2.2. Contracciones

Este módulo se encarga de separar una contracción en las palabras que la componen. Por ejemplo, la salida correspondiente a la contracción *you're* es *you are*, es decir, *you're* se ha descompuesto en el pronombre *you* y la conjugación verbal *are*.

2.2.1. Modelamiento del módulo de contracciones

Se puede definir una contracción en términos generales de la siguiente manera:

Wn es una contracción de $W1 W2... Wk$ si $W1 W2... Wk$ se puede reescribir como Wk abreviadamente. Donde W_i es una palabra. Por ejemplo:

you're es una contracción de *you are* ya que *you are* se puede reescribir como *you're* abreviadamente.

3. TOKENIZACIÓN

El prototipo implementado para resolver el problema de tokenización acepta una gran cantidad de patrones. Aunque se hizo uso de la librería tokenize de *NLTK LITE*¹, son las expresiones regulares implementadas las que realizan la tarea de tokenización.

3.1. Modelamiento del tokenizador

Los tokens son conformados por la asociación de

diferentes caracteres los cuales cumplen unas reglas específicas. Para cada tipo de asociación entre caracteres se pueden definir reglas que representan la manera en como se constituyen. Dichas reglas se pueden expresar a través de expresiones regulares y en algunos casos son dependientes del idioma (aunque en la mayoría de los idiomas son similares). De igual forma deben ser definidas de manera jerárquica de tal manera que un token con una regla compleja no sea interpretado como varios tokens expresados a través de reglas sencillas. Así, el orden que deben tener los tokens de acuerdo a la jerarquía es: fechas, horas, números, cantidades de porcentaje, valores monetarios, palabras, signos de puntuación.

A continuación se muestra el resultado de aplicar en un texto de entrada el tokenizador explicado anteriormente:

```
This is a word
UB40 Windows98 350AC
alpha-galactosyl-1,4-beta-galactosyl-specific Zig-Zag
Finland's capital
$2.500 $1.234.456.450,23
2.12123123 12123123.00000 25%
05/09/2005 5 de Agosto de 2005 05-09-2005
13:10 5:15 a.m. 5:15 P.M.
Mr. Sebastian Marin, Mr. Jose Ferney Franco and Mr. Cesar
Fredy Gil develop a tesis using the dotproject developer
- ? () ,
New York
```

Se obtiene la siguiente salida:

```
['This', 'is', 'a', 'word', 'UB40', 'Windows98', '350', 'AC',
'alpha-galactosyl-1,4-beta-galactosyl-specific', 'Zig-Zag',
'Finland's', 'capital', '$2.500', '$1.234.456.450,23',
'2.12123123', '12123123.00000', '25%', '05/09/2005', '5',
'de', 'Agosto', 'de', '2005', '05-09-2005', '13:10', '5:15 a.m.',
'5:15 P.M.', ',', 'mr.', 'Sebastian', 'Marin', ',', 'mr.', 'Jose',
'Ferney', 'Franco', 'and', 'mr.', 'Cesar', 'Fredy', 'Gil', 'develop',
'a', 'tesis', 'using', 'the', 'dotproject', 'developer', '-', '?', '(', ')', ',',
'New York']
```

Se puede observar como la mayoría de los problemas que se pueden presentar en la tokenización para el idioma inglés se resuelven con este prototipo. Además a través de la creación de un archivo de excepciones se pueden tokenizar palabras que no sean abarcadas por las expresiones regulares, siendo el caso de palabras compuestas y abreviaciones como: Mr., Dr., m.p.h., New York, etc.

4. DICCIONARIO

La estructura con que cuenta el diccionario tiene grandes ventajas en cuanto al acceso, interpretación y fácil búsqueda de de información de tipo sintáctico y semántico. Se optó por emplear los arreglos asociativos y no otra estructura como una base de datos ya que comparativamente con esta no es necesario tener un "pesado" motor corriendo. Los arreglos asociativos ofrecen una estructura muy semejante a la de los diccionarios de papel, con los cuales los usuarios están normalmente familiarizados. La edición del diccionario

¹ Para mayor información de cómo usar e instalar NLTK <http://nltk.sourceforge.net/>

se hace directamente en un archivo de texto, permitiendo una fácil actualización y brindando una excelente escalabilidad.

4.1. Modelamiento del Diccionario

Para el diccionario se optó por emplear las tablas hash. La estructura de un diccionario de papel es muy similar a una tabla hash ya que este se encuentra indexado por palabras, las palabras tienen así mismo índices indicando las diferentes categorías de los significados de las palabras y dichas categorías y tienen asociados múltiples roles, por último cada rol tiene un significado y opcionalmente, en caso de ser un sustantivo, posee información de género. El modelo del diccionario se observa en la Figura 1.

4.2. Formato de Entrada Y Salida

Haciendo uso de los arreglos asociativos de Python se carga el diccionario, la entrada de este se lee desde un archivo de caracteres que en cada línea tiene una palabra y su correspondiente significado. El formato de cada línea del archivo de diccionario es:

palabra: categoria1. [rol1:] significado [; rol2: significado] ...] [; categoria2 . [rol1:] significado [; rol2 : significado] ...] ...

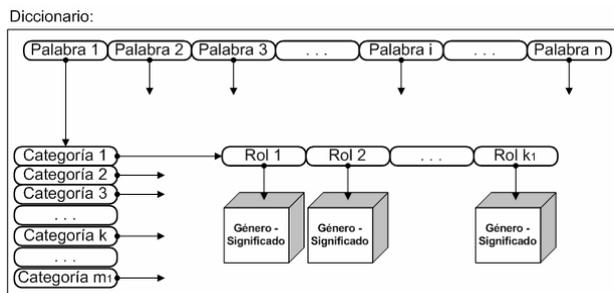


Figura 1 Estructura del diccionario

Entonces, para las siguientes palabras la entrada del diccionario sería:

cry: v. llorar; n. llanto
 play: v. Dep: jugar + Teat: actuar + Mus: tocar; n. Dep: juego + Teat: obra

Por **categoría** se entiende la **clasificación sintáctica** de la palabra, es decir si la palabra es un verbo (*v*), sustantivo (*n*), adjetivo (*adj*), etc. Por **rol** se entiende la **clasificación semántica** de la palabra, por ejemplo: *Mús* → *Música*, *Zool* → *Zoología*.

Las categorías son obligatorias, los roles son opcionales, en caso de no haber un rol en un significado de una palabra, se usa el rol 'NOROL'.

El arreglo asociativo creado tiene el siguiente formato:

```
{'palabra1': {'categoria1': {'roll': 'Sig'}, 'categoria2': {'roll': 'Sig'}, ...}, 'palabra2': {'categoria1': {'roll': 'Sig'}, 'categoria2': {'rol': 'Sig'}, ...}, ...}
```

A continuación se muestra a manera de ejemplo la estructura del diccionario teniendo únicamente tres

entradas (*play, cry e important*):

```
{'play': {'VB': {'Dep': ['jugar'], 'NOROL': ['jugar'], 'Mus': ['tocar'], 'Teat': ['actuar']}, 'NN': {'Dep': ['juego'], 'NOROL': ['juego'], 'Teat': ['obra']}}, 'cry': {'VB': {'NOROL': ['llorar']}, 'NN': {'NOROL': ['llanto']}}, 'important': {'JJ': {'NOROL': ['importante']}} }
```

De esta manera, para realizar la búsqueda de la información referente a una palabra en el diccionario basta con ejecutar una instrucción como la siguiente:

diccionario[palabra][categoria][rol]: obteniendo el significado para la palabra, categoría y rol dados. Siendo así, para encontrar el significado de la palabra *play* bajo la categoría *verbo* y rol *deporte*, se requiere la siguiente instrucción:

```
>>>diccionario['play']['VB']['Dep']  
['jugar']
```

5. ANÁLISIS MORFOLÓGICO

El análisis morfológico toma como entrada los tokens arrojados por el proceso de tokenización, a cada una de las palabras las clasifica morfológicamente. Para cada una de ellas se crea un arreglo asociativo que tiene como llaves las diferentes categorías que puede tener dicha palabra, y asociada a esta llave hay una tupla ordenada (*palabraReal, categoriaReal*) cuyos componentes corresponden a la raíz de la palabra y la categoría de dicha raíz. De esta manera, para un conjunto de palabras se obtiene un arreglo asociativo del siguiente tipo:

```
{'palabra1': {'categoria1': (palabraReal, categoriaReal), 'categoria2': (palabraReal, categoriaReal)...}, 'palabra2': {'categoria1': (palabraReal, categoriaReal), 'categoria2': (palabraReal, categoriaReal), ...}, ...}
```

5.1. Modelamiento del Analizador Morfológico

Si una palabra *w'* puede ser originada a partir de un lexema *w* haciendo uso de una o varias reglas de derivación o inflexión morfológica R_i . Se dice entonces que *w'* se obtiene mediante composición morfológicas a partir de *w*.

$w' = R_i(w)$ Donde *w* es la palabra raíz y *w'* es la palabra obtenida a partir de *w* haciendo uso de la regla R_i . Por ejemplo:

Sea $w = cat$ y R_i la regla de formación de plural. Así pues *w'* será el plural de *cat* dando como resultado que $w' = cats$.

Ahora bien, en el proceso de análisis morfológico es necesario realizar el proceso inverso. Dada una palabra se requiere saber de que palabra se deriva esta.

De tal manera que si se tiene una palabra *w'* y si

aplicando a esta una función R_i^{-1} se obtiene una palabra raíz w (la cual debería estar en el diccionario), se dice entonces que posiblemente² w' es originada a partir de un proceso de inflexión o derivación de w .

Aún teniendo dicho problema de ambigüedad, todas las raíces w posibles así como las reglas R_i aplicadas para llegar a la palabra derivada w' deben ser desplegadas, ya que más adelante el proceso de análisis sintáctico se hará la desambiguación.

Es necesario tener en cuenta que si al aplicar una regla inversa R_i^{-1} a una de las palabras w' (las cuales en este caso son los tokens obtenidos en el proceso de tokenización) se obtiene una raíz w . Es importante verificar que al aplicar la regla R_i se obtenga nuevamente la palabra w' ya que sino es así podría afirmarse que w' se deriva de w a partir de la regla R_i siendo esto incorrecto.

A continuación se muestra el resultado de realizar análisis morfológico a las palabras *move* y *crying*:

Token: 'moves'
 Etiqueta: 'VBZ' (Palabra Real, etiqueta Real): ('move', 'VB')
 Etiqueta: 'NNS' (Palabra Real, etiqueta Real): ('move', 'NN')

Token: 'crying'
 Etiqueta: 'VBG' (Palabra Real, etiqueta Real): ('cry', 'VB')
 Etiqueta: 'NN' (Palabra Real, etiqueta Real): ('crying', 'NN')

Etiqueta: 'JJ' (Palabra Real, etiqueta Real): ('crying', 'JJ')

6. ANÁLISIS SINTÁCTICO

El analizador sintáctico que se ha construido permite generar las diferentes estructuras gramaticales válidas que puede tener el texto de entrada.

6.1. Modelamiento del Analizador Sintáctico

Se ha acogido el enfoque de constituyentes y dentro de este, se usa un *parser* descendente que permite escoger las diferentes oraciones bien formadas del texto.

Las reglas de producción están conformadas por las producciones gramaticales y léxicas. Las producciones gramaticales definen la gramática de oraciones bien formadas del idioma inglés. Las producciones léxicas indican las posibles etiquetas que puede tener cada una de las palabras del texto de entrada, a partir de estas es que el *parser* define las posibles listas de producción de salida (árboles sintácticos), esto es, las posibles estructuras correspondientes a oraciones bien formadas

² Se dice posiblemente ya que en este paso no se tiene certeza que tal palabra realmente corresponda a dicha raíz. Esta información sólo se puede saber más adelante el proceso de análisis sintáctico.

que puede tener el texto de entrada.

A continuación se muestra el resultado de realizar el proceso de análisis sintáctico a una frase simple:

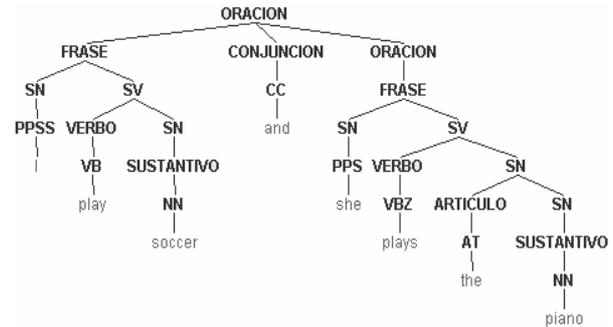


Figura 2 Estructura sintáctica “I play soccer and she plays the piano”

El analizador sintáctico identifica correctamente la estructura gramatical de la oración. Realizando desde la descomposición de sus frases en su sintagma nominal y sintagma verbal hasta llegar a identificar la categoría adecuada para cada palabra.

7. ANÁLISIS SEMÁNTICO

El análisis semántico muestra la funcionalidad de un motor de inferencia de conocimiento semántico el cual es capaz de asociar las clasificaciones comunes entre los componentes de una oración siendo posible resolver problemas de orden semántico.

7.1. Modelamiento del analizador semántico

Para abordar el problema, se hace necesario retomar la construcción básica del idioma la cual dice que la frase es *un sujeto que realiza una acción sobre un objeto*. El sujeto se conoce como sintagma nominal, la acción que se realiza sobre el objeto se conoce como sintagma verbal y este a su vez está compuesto por una acción (parte verbal) realizada sobre un objeto (sintagma nominal). Dicha composición se puede observar en la Figura 3.

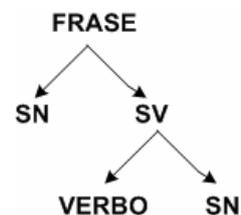


Figura 3 Estructura gramatical básica de una frase

Teniendo en cuenta lo anterior, es necesario encontrar la manera de relacionar los roles existentes en cada una de las partes en que se constituye la frase, el sintagma nominal, la parte del verbo del sintagma verbal y el sintagma nominal del sintagma verbal. El análisis debe ser efectuado al nivel de la frase y debe ser capaz de relacionar adecuadamente la información de restricciones selectivas.

Dadas las anteriores especificaciones, se definen los siguientes conjuntos:

rolesSN = { x | x es un rol de la palabra w en la categoría cat, tal que w pertenece al sintagma nominal de la frase actuando con la categoría cat }

rolesSVVerbo= { x | x es un rol de la palabra w en la categoría cat, tal que w pertenece a la parte del verbo del sintagma verbal de la frase actuando con la categoría cat }

rolesSVSN = { x | x es un rol de la palabra w en la categoría cat, tal que w pertenece a la parte del sintagma nominal del sintagma verbal de la frase actuando con la categoría cat }

A continuación, se muestra el resultado de ejecutar análisis semántico a la frase: I play soccer and she plays the piano. Cuya traducción correcta al español es: yo juego fútbol y ella toca los pianos. En el ejemplo anterior existe ambigüedad semántica al tratar de traducir el texto, el verbo play puede ser tres cosas (tocar, jugar o actuar). Los significados se pueden deducir sabiendo que piano es un instrumento musical y que soccer es un juego.

En la Figura 4 se puede los roles asociados a cada palabra, este árbol se obtiene después de ejecutar una función que agrega los roles semánticos.

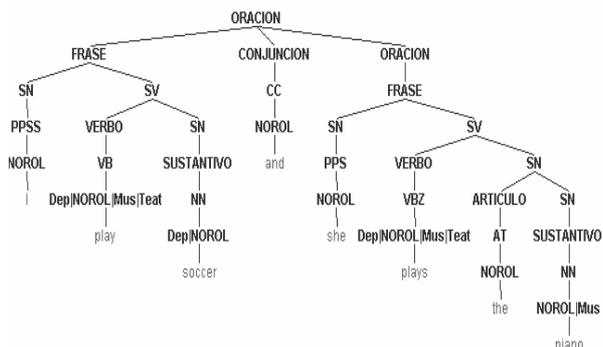


Figura 4 Estructura sintáctica y géneros “I play soccer and she plays the piano”

En la Figura 5 se muestra el árbol de la Figura 4 después de ejecutar una función encargada de eliminar las ambigüedades semánticas.

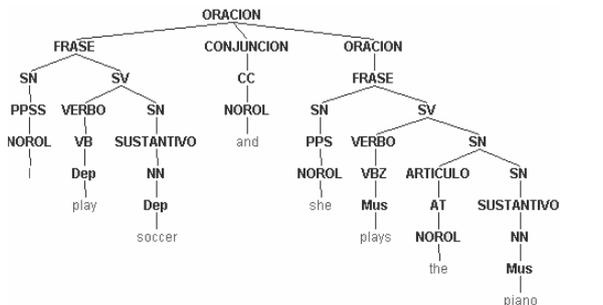


Figura 5 Estructura sintáctica con género no ambiguo “I play soccer and she plays the piano”

8. SÍNTESIS AL LENGUAJE DESTINO

El trabajo de este módulo consiste en analizar las frases

recursivamente, identificando el sintagma nominal y el sintagma verbal, para así seleccionar la conjugación adecuada de acuerdo a la persona y tiempo.

8.1. Modelamiento de la Síntesis al Lenguaje Destino

8.1.1. Conjugador

Este módulo es el encargado de la conjugación de los verbos en los diferentes tiempos para las diferentes personas. De tal forma que:

Vc = Conjugar(Vb, Su, T) dé como resultado el verbo Vc, el cual es el resultado de conjugar el verbo Vb en el tiempo T para la persona Su.

8.1.2. Conformación de plurales

La conformación de plurales, tal como la conjugación de verbos, varía de acuerdo al idioma. En general, un módulo Plural(W) cumple la siguiente condición:

W' = Plural(W) Donde W es la palabra a aplicar el plural y W' es el plural de W .

8.1.3. Cambio de género

Este módulo se aplica para idiomas que manejan género como el español. De tal forma que es necesario modificar el género para ciertas palabras que dependen del género de otras (generalmente adjetivos). Un módulo Genero(W) cumple la siguiente condición:

W' = Genero(W) Donde W es la palabra a la que se quiere cambiar el género (normalmente de masculino a femenino) y W' es la palabra con diferente género de W .

8.1.4. Conjunción de nombres

Un problema complejo y que se presenta en cualquier idioma es la combinación de varios pronombres unidos por una conjunción y realizando una misma acción, es necesario realizar un módulo que a partir de la conjunción de múltiples personas se pueda seleccionar la persona y el número adecuado. Consiste en un módulo que permita realizar la conjunción entre dos personas de la siguiente manera:

Persona_r,numero_r=
persona_numero_conjuncion(persona_1, numero_1,
persona_2, numero_2)

Tal que persona_r, numero_r es el resultado de realizar la conjunción de la persona1, numero_1 con la persona_2, número_2. Donde persona_i, numero_i son la persona y número (plural o singular) de la persona.

En la Tabla 1 se observan las entradas y salidas para algunos ejemplos en el proceso de síntesis.

Entrada	Salida
I play soccer and you play piano	Yo juego fútbol y tu tocas piano

I play soccer and she plays the piano	Yo juego fútbol y ella toca los piano
Sebastian works with Tokenization	Sebastian trabaja con Tokenization
I'm working with Tokenization	Yo estoy trabajando con tokenization
Jose Ferney Franco fishes in the river	Jose Ferney Franco pesca en el río
I button the button	Yo abotono el botón
The computer is an electronic device	El computador es un dispositivo electrónico
The kid plays with the pianos and the girl plays with dolls	El niño toca con los pianos y la niña juega con muñecas
The kid bleats and the kid speaks	El cabrito bala y el niño habla
The kid plays piano	El niño toca piano
The kid plays with the girl	El niño juega con la niña
I work tokenization with Sebastian	Yo trabajo tokenization con Sebastian
New York is a city	Nueva York es una ciudad

Tabla 1 Entrada y salida del proceso de síntesis

En el ejemplo anterior se puede observar como el conjugador realiza su tarea, nótese como para el ejemplo *I play soccer and you play piano* se identifica que la primera aparición de *play* debe ser conjugado en primera persona, en español *jugar* quedaría *juego*. La segunda aparición debe ser conjugada en segunda persona, es español el verbo *tocar* quedaría *tocas*.

9. CONCLUSIONES

En este trabajo se utilizaron las etapas del proceso de traducción para producir traducciones de oraciones afirmativas en presente simple del inglés al español, resolviendo ambigüedades semánticas por medio de la información suministrada por el diccionario, el cual le brinda la información semántica necesaria para que el analizador asocie los roles adecuados en las oraciones. Si el sintagma nominal y el verbal tienen roles semánticos en común, los roles restantes son eliminados y así la síntesis al lenguaje destino puede hacer la transferencia sin ambigüedades. El motor de inferencia base para este proceso es un aporte que puede ser usado en trabajos futuros de tal manera que se logren avances considerables en el área de la traducción automática.

Cada etapa del proceso de traducción automática aquí planteada puede ser mejorada en cuanto a su alcance y eficiencia, podría pensarse en realizar en análisis semántico con inteligencia artificial u otro método.

10. BIBLIOGRAFÍA

[1] American Demographics. 1 Jan 2005. <http://www.demographics.com/publications/fc/97%5Ffc/9712%5Ffc/f971201.htm>

- [2] ARNOLD, Doug D. A. "Machine Translation: an Introductory Guide." NCC BlackWell (1994).
- [3] Global Internet Statics. 7 Feb 2006. <<http://www.glgreach.com/globstats/index.php3>>
- [4] GREFENSTETTE, Gregory G. G. y TAPANAINEN, Pasi P. T. 1994. What is a Word, What is a sentence? Problems of Tokenization.
- [5] HANCOX, Peter P. H. "MORPHOLOGICAL ANALYSIS: THE CLASSIFICATION OF [20] [20]
- [6] HERNANDEZ, Pilar P. H. "En torno a la traducción automática." El concepto de la traducción automática (2002).
- [7] HUTCHINS, John J. H. et al. EUROPEAN ASSOCIATION FOR MACHINE TRANSLATION, Compendium of Translation Software. 2005.
- [8] HUTCHINS, John J. H. Machine translation: Computer-based translation.
- [9] HUTCHINS, John. "The State of Machine Translation in Europe and Future Prospects". 16 Mar 2005. <<http://gda.utp.edu.co/traductor/Documentacion/EstadoArte/Articulos/articulo5.htm>>
- [10] MERRIMACK, N. H. M. "Enterprise Translation Server." Overview: How real-time translation works.
- [11] PATTERSON, Wright W. P. Civil Engineering Center, Estimaciones de costos de traducciones. 1992.
- [12] SAG, IVAN A. and WASOW, TOMAS. Syntactic Theory: A Formal Introduction.
- [13] The Apostrophe. 28 Jan 2006. <http://owl.english.purdue.edu/handouts/grammar/g_apost.html>
- [14] Warren Weaver Memorandum. MT News International Jul 1999.