Sistema Incremental Generador de Oraciones y de Descodificación Lingüística.

José Luciano Maldonado.
luzmalvy@telcel.net.ve
maldonaj@faces.ula.ve

Resumen: se describe la implementación experimental de un sistema de computación para crear modelos de contextos. El procedimiento es altamentedeterminístico y los modelos se crean a partir de un conjunto de oraciones y párrafos propios de los contextos. Los modelos obtenidos capturan historias de palabras presentes en las oraciones de entrenamiento, por lo que permiten generar automáticamente oraciones asociadas al tema que modelan, y hacer reconocimiento de frases y oraciones.
Estos modelos tienen carácter adaptativo e incremental, pues, se pueden re-ajustar con nuevas oraciones y párrafos sin perder la capacidad lograda en el ajuste previo, por el contrario se mejora su rendimiento.
Estamos describiendo entonces, un tipo dedescodificador lingüístico que podría formar parte de un sistema de reconocimiento automático de la voz, que le agrega a la propiedad de éste, de reconocer secuencias de palabras, la propiedad de revisar si esas secuencias constituyen oraciones válidas de acuerdo a un contexto.

1.- Introducción.

El creciente e indetenible desarrollo de las computadoras de altísima velocidad de procesamiento de información y de gran capacidad de memoria principal que observamos en la actualidad, nos lleva a pensar que puede ser posible el diseño y construcción de sistemas automáticos de reconocimiento del habla cuyo módulo descodificador linguístico comprenda un modelado de lenguaje que copie y codifique todos los componentes gramaticales de un corpus de entrenamiento. Por nuestro lado, como parte de un intento por incursionar en este apasionante mundo del Reconocimiento Automático del Habla, hemos desarrollado un sistema (un conjunto de programas de computación), a través del cual hemos observado que partiendo del modelo de un pequeño corpus compuesto de un conjunto de oraciones de un contexto particular, se puede generar automáticamente una gran cantidad de oraciones gramaticalmente válidas respecto a ese contexto. También, nuestro sistema en pruebas de descodificación lingüística rechaza, como fuera de contexto o incorrectas gramaticalmente, aquellas secuencias de palabras que contengan palabras que no están presentes en su vocabulario o que contenga historias que no tiene codificadas en su memoria.
Pensamos, que un tipo de sistema de descodificación y tratamiento de la información como el que presentamos en este trabajo, puede cubrir de manera exitosa el reconocimiento en diversos contextos donde el tamaño del vocabulario sea de varios miles de palabras.
La descripción del sistema comprende una primera sección en la que se presentan algunos términos que emplearemos a lo largo del informe, una sección dedicada a explicar la implementación, entrenamiento y funcionamiento del generador de modelos de contextos, una sección donde se explica el funcionamiento del generador de oraciones, una sección dedicada a mostrar como trabaja el reconocedor o descodificador linguístico, una sección donde se indican las pruebas realizadas, se señalan los resultados obtenidos y se enumeran algunas conclusiones a las que hemos llegado.

2.- Terminología utilizada.

Corpus de entrenamiento: El conjunto formado por las oraciones y párrafos a partir del cual se construye el modelo del contexto

tanto para el reconocimiento como para la generación de las oraciones.

Vocabulario: el vocabulario comprende el conjunto de palabras distintas que se encuentran en el corpus de entrenamiento.

Entrenamiento: proceso mediante el cual se crean los modelos de los contextos.

Historias: conjuntos de palabras que aparecen en forma contigua en el corpus de entrenamiento, [1].

Ejemplo de historias: sea la siguiente, una oración presente en el corpus de entrenamiento:

"tres razones parecen ser el origen de este hecho".

Una historia de dos palabras sería: "el origen".

Una historia de tres palabras sería: "el origen de".

Contexto: área del conocimiento a la cual pertenecen las oraciones y párrafos que componen el corpus de entrenamiento.

Reconocimiento: el proceso de descodificación de secuencias de palabras que se le presentan al sistema, a través del cual se determina si dicha secuencia es una oración válida respecto a las reglas gramaticales presentes en el contexto que se modela.

Generación de oraciones: proceso mediante el cual se crea una oración a partir del modelo del contexto.

Oración gramáticalmente válida: oración que tiene una estructura que sigue las reglas gramaticales encontradas en el corpus de entrenamiento.

Hipótesis de oraciones: conjunto de posibles oraciones a las que podría corresponder una secuencia de palabras a reconocer o a generar.

3.- Generador de modelos de contextos.

En la figura 1 se muestran los elementos principales del sistema, sus entradas y salidas, y se da una idea gráfica de cómo interactuan dichos elementos.

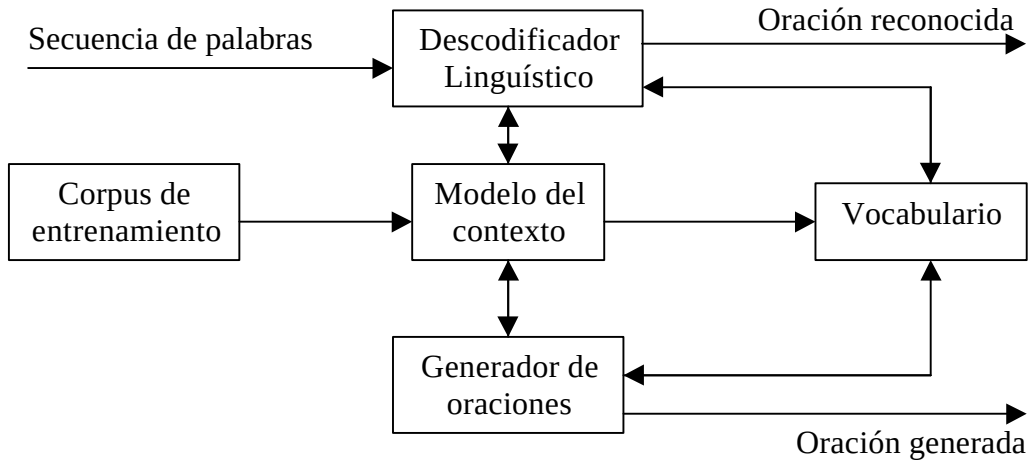

Figura 1. Estructura del sistema.

Para crear un modelo partimos de un conjunto gramaticalmente correcto de oraciones y párrafos propio del contexto que se quiere modelar. Se modela el contexto, aunque creemos que si se alimenta el modelo en forma incremental se podrá llegar a modelar en buena forma el lenguaje al cual pertenece el corpus de entrenamiento. El entrenamiento o modelado del contexto consiste básicamente en la búsqueda, codificación y almacenamiento de la ocurrencia de historias de palabras contiguas dentro de las oraciones y párrafos del corpus de entrenamiento. Creamos bloques codificados de historias de palabras. Las palabras presentes en las historias se codifican a través de números enteros. En la figura 2, se da una idea de cómo se ubican las historias presentes en el corpus de entrenamiento. Allí, podemos ver que la historia "el origen" se codifica como la secuencia de enteros " 5 6" y la historia "el origen de" se codifica como la secuencia de enteros "5 6 7". El número entero lo asignamos de acuerdo a la posición que tiene

cada una de las palabras en el vocabulario obtenido del corpus.
A la primera palabra que aparece en el primer corpus de entrenamiento (tengamos presente que se puede trabajar de manera incremental con varios corpus) se le asigna el número 1, luego, a la siguiente palabra distinta a la primera, se le asigna el número 2 y así sucesivamente. Entonces, a la n-ésima palabra distinta del corpus, se le asigna el número n.
Se forman los bloques siguientes:
Un bloque que contiene una lista codificada de las palabras que en el corpus inician las oraciones; éste bloque es usado por el generador de oraciones y lo llamaremos **bloque1**.
Un bloque que contiene una lista de historias de dos palabras. Este bloque puede ser usado tanto por el generador de oraciones como por el descodificador linguístico. A éste lo llamaremos **bloque2**.
Un bloque que contiene una lista de historias de tres de palabras, igual que el bloque2, puede ser usado tanto por el generador como por el descodificador. Este bloque comprende las tripletas de palabras presentes en el corpus de entrenamiento. Lo llamaremos, **bloque3**.

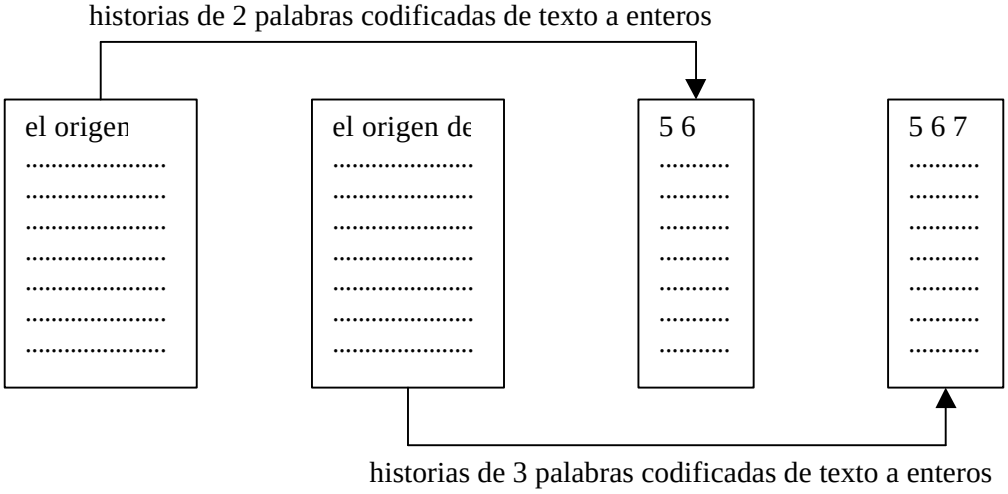

Figura 2: Ejemplos de bloques de historias tomadas del corpus de entrenamiento

Un bloque que contiene una lista de historias de tres palabras como el bloque3, pero a diferencia de aquel, cada historia contiene las tres últimas palabras de cada oración y párrafo del corpus de entrenamiento. Este bloque le permite al generador de oraciones darse cuenta de cuando ha de finalizar la producción de una oración. Este es el **bloque4**.
La codificación del corpus de entrenamiento en los bloques 1, 2, 3 y 4, se hace para facilitar la creación de las hipótesis de las oraciones para el reconocimientolinguístico, para la generación de oraciones y para agilizar el manejo de los datos.
Una vez que se crean y se codifican los bloques antes mencionados se desechan ciertas historias para bajar un poco la redundancia en la codificación, y para agilizar el trabajo de los programas que reconocen y generan oraciones. La eliminación de historias es un tanto arbitraria; por ejemplo, se hicieron pruebas donde se eliminan por un lado, las historias de dos palabras cuya frecuencia de aparición en el corpus es baja, y por otro lado, almacenamos las historias de tres palabras que se inician con las historias de dos palabras eliminadas.
Por ejemplo, supongamos que en el corpus el par de palabras "la linguística" aparece 7 veces y el par "sujeto a" aparece 1 vez. Entonces, almacenamos el primer par, mientras que el segundo par no, pero en su

lugar almacenamos tripletas como "sujeto a condiciones".

Trabajando de esta manera, esperamos asegurar que aquellas historias que aparecen con baja frecuencia también sean tomadas en cuenta a la hora de hacer reconocimiento y generación de oraciones. Está claro que al dejar por fuera muchas historias de tres palabras el rendimiento del sistema disminuye. Sin embargo, pensamos que tenemos varias alternativas: una sería codificar y almacenar cada una de las historias que aparecen en el corpus, otra sería agregar más oraciones al corpus donde aparezcan con más frecuencia las historias antes rechazadas, es decir re-entrenando el modelo. La segunda alternativa es la que hemos usado, puesto que el modelo que se crea de esta manera es adaptativo e incremental debido a que se puede repetir el procedimiento con otro conjunto de entrenamiento del contexto, y re-ajustar el modelo sin perder la información obtenida en un ajuste previo. Esto permite que en forma incremental se pueda enriquecer no sólo el vocabulario del módelo, sino que se puede aumentar también su capacidad para generar oraciones y para hacer el reconocimiento.

El proceso de re-entrenamiento o re-ajuste consiste en que una vez que tenemos un modelo creado a partir de un corpus que llamaremos Corpus1, podemos seleccionar otras oraciones del mismo contexto, pero diferentes a las de dicho corpus, y elaborar con éstas un Corpus2. Utilizando este nuevo corpus codificamos todas sus historias y sumamos la frecuencia de aparición de aquellas que aparecen tanto en el Corpus1 como en Corpus2, por lo que tendremos una frecuencia acumulada para esas historias. Finalmente se almacena en los bloques mencionados 1, 2, 3 y 4, las historias pares de mayor frecuencia, junto con las historias triples que contienen como sus dos palabras iniciales aquellas que constituyen las historias pares eliminadas. Podemos ver que se trata de una extensión del modelo creado con el Corpus1, al que estamos agregando nuevas historias y hasta nuevas palabras a su vocabulario, puesto que en el segundo corpus, pueden aparecer no sólo historias que no estaban presentes en el primero sino también nuevas palabras. Este proceso se puede repetir con tantos corpus como consideremos necesario para nuestras aplicaciones.

4.- Generador de oraciones.

A partir del modelo codificado como se explicó en la sección anterior, el sistema es capaz de producir aleatoriamente oraciones dentro del contexto que se modela.

El algorítmo para generar una oración es el siguiente:

1.- Se escoge en forma aleatoria una palabra de la lista del bloque1, digamos w_1, y la mostramos en la pantalla del sistema.

Por ejemplo, se escoge del bloque1, la palabra "Esta".

2.- Del bloque3, se agrupan aquellas historias que se inician con la palabra w_1 en un bloque nuevo, el **sub-bloque31**.

Por ejemplo, "Esta especificación debe", "Esta oración declarativa", "Esta oración corresponde", "Esta distinción permite", "Esta juega un", etc.

3.- Si el sub-bloque31 no está vacío, se escoge aleatoriamente una de las historias que contiene y se muestran en la pantalla las dos palabras que siguen a w_1 en esa historia.

Por ejemplo, se escoge la historia "Esta oración corresponde" y se muestra en pantalla las palabras "oración corresponde".

4.- Se continúa la búsqueda de historias en el bloque2. Las historias que interesan de este bloque son aquellas que tienen como palabra inicial la última que aparece en la historia seleccionada del sub-bloque31 cuando este bloque no está vacío, digamos historias que comiencen con w_3. Cuando el sub-bloque31 está vacío las historias que interesan son aquellas que tienen como palabra inicial a w_1. Se forma así un nuevo bloque, el **sub-bloque21**.

Por ejemplo, "corresponde a". En este ejemplo, el sub-bloque21 está constituido por una sola historia.

5.- Si el sub-bloque21 es no vacío, se selecciona aleatoriamente una de sus historias y se muestra la segunda palabra de esa historia.

Por ejemplo, se selecciona la historia "corresponde a" porque es la única, pero en caso de haber más de una, se selecciona una de ellas aleatoriamente. Se muestra la segunda palabra, "a". Hasta este momento tenemos la frase "ESTA ORACION CORRESPONDE A".

6.- Se continua la búsqueda en el bloque4 que contiene historias que finalizan oraciones. Las historias que interesan de este bloque son aquellas que comienzan con la última palabra de la última historia seleccionada en los pasos previos. Se forma el **sub-bloque41**.

Por ejemplo, las historias que interesan de este bloque serían las tripletas que comienzan con "a". En este ejemplo, no encontramos en el bloque4 historias que comiencen con "a". Es decir que el sub-bloque41 es vacío.

7.- Si el sub-bloque41 es no vacío, entonces se selecciona aleatoriamente una historia y se muestran sus dos últimas palabras. Aquí finaliza la generación de una oración.

8.- Se actualiza w_1 con el código de la palabra con la que finaliza la última historia seleccionada.

Para el ejemplo, que se presenta aquí, w_1 tomaría el índice de "a".

9.- Volvemos al paso 2.

Para el ejemplo, después de volver al paso 2 y encontrar una historia en el bloque4, se obtiene la oración "ESTA ORACION CORRESPONDE A UN RITMO SILABICO".

La mayoría de las oraciones que se generan de esta manera no aparecen en el (los) corpus de entrenamiento, es decir, se forman a través de la conexión adecuada de las historias codificadas.

La oración que mostramos en el ejemplo no aparece en el corpus, pero si aparecen las oraciones que mostramos a continuación y que por sus componentes podemos darnos cuenta que intervienen mucho en la producción de la oración mencionada.

a.- ESTE TIPO DE ORACION CORRESPONDE A UN ENUNCIADO NEUTRO DESPROVISTO DE ASPECTOS EXPRESIVOS Y APELATIVOS ESPECIALES.

b.- CORRESPONDE A LA ULTIMA SILABA PORTADORA DE ACENTO LEXICO EN EL GRUPO MELODICO.

c.- ESTA ORACION DECLARATIVA ESTA CONSTITUIDA POR TRES UNIDADES TONALES.

En resumen, la generación de una oración, se hace entonces con una búsqueda sucesiva de palabras en las historias de los bloque3, bloque2 y bloque4.

El proceso finaliza cuando se encuentra al menos una historia en el bloque final (bloque4) o cuando no aparece ninguna historia en ninguno de los tres bloques que pueda continuar a una precedente. Este caso se puede presentar cuando se selecciona una historia que en el corpus está ubicada al final de una oración, estas historias con frecuencia finalizan con palabras que no forman otras historias por lo tanto ninguna historia las podrá seguir.

En este trabajo, se diseñó y construyó (por programación) el generador de oraciones descrito, con el fin de tener una idea de las secuencias de palabras que podría reconocer el descodificador linguístico que posteriormente desarrollaríamos y que describiremos en la próxima sección. En este momento podemos suponer que el modelo de contexto es una red de historias codificadas que contiene las oraciones que pueden ser reconocidas por eldescodificador linguístico. La utilidad del generador de oraciones en este trabajo fue prevista sólo para mostrar las oraciones presentes en el modelo y que por lo tanto pueden ser reconocidas.

5.- Reconocedor o descodificador linguístico.

La parte de un reconocedor del habla que convierte los datos acústicos de una pronunciación en una secuencia de símbolos linguísticos (por ejemplo, una secuencia de fonos, una secuencia de palabras, etc.) se llama Descodificador Acústico, mientras que el Descodificador Linguístico es la parte del reconocedor del habla que determina si esa secuencia de símbolos corresponde a una oración válida de un lenguaje [2].

Tal como se aprecia en la figura 1, en este trabajo sólo se desarrolla el descodificador linguístico por lo que sus pruebas se realizan suponiendo que de existir un descodificador acústico, recibiría de éste una secuencia de palabras.

El procedimiento a través del cual el sistema puede reconocer una secuencia de palabras (w_1, w_2, ..., w_n) [2] como gramaticalmente correcta a partir del modelo de contexto, se presenta a continuación:

1.- Recibe la primera palabra de la secuencia, w_1.

2.- Averigua si w_1 está presente en el vocabulario. Si w_1 no forma parte del vocabulario, la rechaza y por lo tanto a la secuencia por estar fuera del contexto. Si pertenece al vocabulario muestra w_1 en la pantalla.

3.-Busca historias que se inicien con w_1 en bloque2 y bloque3. De esta manera se generan dos nuevos bloques de posibles partes de oraciones que de acuerdo al lenguaje que modelan, podrían formarse partiendo de w_1. Uno de esos bloques es producto de la búsqueda en bloque2, llamémoslo **bloque21** y otro, producto de la búsqueda en bloque3, el **bloque31**. Se generan de esta manera, hipótesis parciales de oraciones. Esto constituye, creemos, una forma para agilizar el proceso, puesto que la búsqueda de la palabra siguiente, w_2, de la secuencia a reconocer se haría solo en bloque21 y bloque31.

Puede ocurrir que no se encuentren historias que se inicien con w_1, es decir, puede ocurrir que el bloque21 y el bloque31 resulten vacíos. En este caso, en el modelo no hay palabras que puedan seguir a w_1 por lo que el reconocedor no admitirá la oración (la secuencia de palabras) y finalizará el reconocimiento.

4.- Se recibe la siguiente palabra de la secuencia, w_2. Si forma parte del vocabulario, entonces se busca su ocurrencia en las historias presentes en el bloque21 y en el bloque31, en caso contrario se rechaza la secuencia.

5.- Si el bloque21 ó el bloque31 son no vacíos, se descartan de estos bloques aquellas historias que no contengan a w_2 después de w_1. Si quedan historias que contengan a w_2 siguiendo a w_1 se muestra w_2 en pantalla, en caso contrario se rechaza la secuencia y se termina su reconocimiento. Esto puede ocurrir, debido a que el par (w_1 w_2) no aparece en el corpus de entrenamiento.

6.- Se vuelve al punto 3, trabajando con w_2 en lugar de w_1, es decir, cada vez que la descodificación llega a este punto, se re-inicia el recorrido trabajando con la última palabra tratada en el recorrido previo.

El reconocimiento de la secuencia de palabras tiene dos formas de finalización: una cuando el descodificador la rechaza debido a que según el modelo no es válida o porque no pertenece al contexto y otra, cuando se recibe el símbolo $ que es el indicador de fin de oración, en este último caso tendremos una oración gramaticalmente correcta.

Por ejemplo, el descodificador linguístico reconocería como válida la secuencia "esta oración corresponde a un rítmo silábico $" (suponiendo que dicha secuencia la recibe desde un descodificador acústico cuya salida

sean palabras), puesto que hemos visto que el generador puede producir dicha oración.
Se puede dar el caso de que se rechacen secuencias que pertenezcan al contexto y que sean gramaticalmente válidas. Esto se puede superar re-entrenando el modelo con nuevos corpus del mismo contexto.
Se puede apreciar que el reconocedor revisa si la secuencia de palabras que recibe es correcta desde el punto de vista de las reglas gramaticales del lenguaje al cual está asociado el contexto y determina también, si forma parte del contexto que se modela.
Aunque los modelos de contextos descritos pueden pensarse como una combinación de Bigramas y Trigramas, en este trabajo no podemos hablar de modelos n-gramas estocásticos, ni de autómatas de estados finitos estocásticos[2], puesto que para la forma como se hace la descodificación no se utilizan las probabilidades. De hecho, este descodificador no mide la probabilidad de que las secuencias estén modeladas o no, simplemente si puede formar una oración que esté en el modelo la acepta de lo contrario la rechaza.

6.- Pruebas.

Los ensayos que se hicieron consistieron en:
1.- Se realizó una prueba inicial con un corpus formado por 160 oraciones y párrafos de distintas longitudes. Se trabajó con longitudes de entre tres y sesenta y tres palabras. Las oraciones fueron tomadas de un texto propio de la lingüística. El corpus completo comprendía 2563 palabras.
2.- Se obtuvo el vocabulario que manejarían tanto el módulo reconocedor como el módulo generador. El vocabulario al principio fue de 816 palabras.
3.- Se buscaron en el texto, el bloque1, el bloque2, el bloque3 y el bloque4 que constituyen el modelo del contexto. Este proceso tuvo una duración aproximada de dos horas en una máquina PC Pentium a 133 Mhz.
4.- Se generaron bloques de oraciones. Este proceso se repitió unas 30 veces. Cada bloque generado comprendía diez oraciones.
5.- Se realizó el reconocimiento de oraciones. La prueba consistió en que dadas secuencias de palabras, se averiguaba si dicha secuencias podían ser reconocidas usando el modelo del contexto.
6.- Se tomaron de nuevo, pequeños corpus de 10 y 20 oraciones y se repitió el procedimiento.

7.- Resultados.

1.- Se pueden utilizar nuevos corpus en forma incremental sin que se pierda la información codificada en ensayos anteriores.
2.- Las oraciones generadas, son en general, más pequeñas en longitud respecto a las contenidas en el corpus de entrenamiento.
3.- El número de oraciones generadas depende de la longitud del corpus de entrenamiento.
4.- El número de oraciones generadas que son válidas en cuanto a la gramática y al contexto supera el 70% del total de las que se generaron en los ensayos.
5.- El número de oraciones reconocidas que son gramaticalmente correctas y que pertenecen al contexto es cercano al 90%, cuando esas oraciones se escogen muy parecidas a las del corpus de entrenamiento.
6.- Cerca del 90% de las oraciones generadas no pertenecen al corpus de entrenamiento, a excepción de algunas oraciones cortas, de tres, cuatro y hasta cinco palabras.
7.- No es posible reconocer todas las oraciones y párrafos, tal como aparecen en el corpus de entrenamiento.

8.- Conclusiones.

En forma incremental se puede lograr cada vez mejor robustés tanto del módulo reconocedor como del generador de oraciones. Claro está, esto conlleva lentitud durante el re-ajuste del modelo, que dependerá de la aplicación y de la máquina.
Como se pueden generar grandes cantidades de oraciones, se puede también reconocer un número grande de frases y oraciones.
Debido a la gran cantidad de oraciones que se pueden generar y que no pertenecen al corpus de entrenamiento, es posible, también reconocer una gran cantidad de oraciones y frases no necesariamente propias del contexto que se modela, pero si propias del lenguaje en el que está escrito el corpus.
No es posible reconocer todas las oraciones y párrafos, tal como aparecen en el corpus de entrenamiento debido a que en la memoria del sistema no aparecen todas las historias presentes en el corpus. Lo que

podría superarse si se almacenan todas las historias que aparecen en el texto, pero esto conllevaría a que la búsqueda tanto en reconocimiento como en generación fuese más lenta.

Trabajando con la codificación y los tamaños de las historias que hemos indicado, se puede crear modelos aceptables de contextos.

Se trata de un reconocedor, un generador de oraciones y un modelador de lenguaje, altamente determinístico.

Se puede hablar de modelar un lenguaje, porque creemos que el modelado de contextos, de esta manera, puede extrapolarse al lenguaje al cual están asociados los contextos.

Este tipo de descodificador linguístico podría funcionar en aplicaciones de reconocimiento donde el tamaño del vocabulario abarca varios miles de palabras.

9.- Referencias.

1.- A. Bonafonte and J. Mariño, "Language Modeling using X-Grams", International Conference on Spoken Language Processing, ICSLP-96.

2.- J. Deller, J. Proakis and J. Hansen, Discrete-Time Processing of Speech Signals. Macmillan Publishing Company.