ARBOR Ciencia, Pensamiento y Cultura
Vol. 190-768, julio-agosto 2014, a160 | ISSN-L: 0210-1963
doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

VARIA / *VARIA*

ESTEGANOGRAFÍA LINGÜÍSTICA EN LENGUA ESPAÑOLA BASADA EN MODELO N-GRAM Y LEY DE ZIPF

SPANISH LINGUISTIC STEGANOGRAPHY BASED ON N-GRAM MODEL AND ZIPF LAW

Alfonso Muñoz Muñoz
amunoz@diatel.upm.es
Universidad Politécnica de Madrid

Irina Argüelles Álvarez
irina@euitt.upm.es
Universidad Politécnica de Madrid

Cómo citar este artículo/Citation: Muñoz Muñoz, A. y Argüelles Álvarez, I. (2014). "Esteganografía lingüística en lengua española basada en modelo N-gram y ley de Zipf". *Arbor*, 190 (768): a160. doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

Recibido: 8 diciembre 2012. Aceptado: 9 julio 2014.

RESUMEN: La esteganografía lingüística es una ciencia que se aprovecha de la lingüística computacional para diseñar sistemas útiles en la protección y la privacidad de las comunicaciones digitales y en el marcado digital de textos. En los últimos años se han documentado múltiples formas de alcanzar este objetivo. En este artículo se analiza la posibilidad de generar automáticamente textos en lenguaje natural en lengua española que oculten una información dada. Se proponen una serie de hipótesis y se experimenta mediante la implementación de un algoritmo. Las pruebas realizadas indican que es factible utilizar modelos N-Gram y peculiaridades derivadas de la ley de Zipf para generar estegotextos con una calidad lingüística tal que un lector humano podría no diferenciarlo de otro texto auténtico. Los estegotextos obtenidos permitirán la ocultación de al menos 0,5 bits por palabra generada.

PALABRAS CLAVE: Esteganografía lingüística; generación automática de estegotextos; N-Gram; algoritmo.

ABSTRACT: Linguistic Steganography is a science that utilises computational linguistics to design systems that can be used to protect and ensure the privacy of digital communications and for the digital marking of texts. Various proposed ways of achieving this goal have been documented in recent years. This paper analyses the possibility of generating natural language texts in Spanish that conceal information automatically. A number of hypotheses are put forward and tested using an algorithm. Experimental evidence suggests that it is feasible to use N-gram models and specific features of the Zipf law to generate stegotexts with a good linguistic quality where human readers could not differentiate the stegotext from authentic texts. The stegotexts obtained allow the concealment of at least 0.5 bits per word generated.

KEYWORDS: Linguistic steganography; automatic generation of stegotexts; N-gram; algorithm.

1. ESTEGANOGRAFÍA LINGÜÍSTICA. LA OCULTACIÓN DE INFORMACIÓN EN LENGUAJE NATURAL

La esteganografía lingüística es una ciencia, variante de la esteganografía[1], que se define como aquel conjunto de algoritmos robustos que permiten la ocultación de información, típicamente binaria, utilizando textos en lenguaje natural como tapadera, los denominados estegotextos. Esta ciencia utiliza principios de la ciencia de la esteganografía e incorpora recursos y métodos de la lingüística computacional como el análisis automático del contenido textual, la generación automática de textos, el análisis morfosintáctico, la lexicografía computacional, las descripciones ontológicas, etc., para crear procedimientos públicos no triviales según los principios de Kerckhoffs (Kerckhoffs, 1883). Es decir, la seguridad de estos algoritmos dependerá exclusivamente de una pequeña información secreta, una clave, compartida por el emisor y el receptor. El algoritmo de ocultación será público, es decir, conocido por todos, incluso por el potencial analista, y los estegotextos resultantes deberán ser resistentes a ataques estadísticos y lingüísticos (coherencia, estructura gramatical, etc.) tanto por software automatizado como por analistas humanos.

El interés en la última década en esta ciencia se debe a que puede dar solución a dos problemas comunes: la privacidad y el anonimato y el marcado digital de textos. Para ello se utilizan dos grandes familias de algoritmos clasificados en técnicas de modificación de textos existentes y en técnicas de generación automática de estegotextos.

Las primeras son las técnicas más tradicionales de ocultación de información que consisten en utilizar un texto existente y ocultar la información mediante la modificación de elementos del mismo. En la última década se han propuesto multitud de técnicas de este tipo (Bergmair, 2007), no exentas de problemas, para diversas lenguas: inglés, español, japonés, chino, árabe, ruso, etc. Aunque estas técnicas se pueden utilizar para la protección y el anonimato de comunicaciones, la comunidad científica tiene más interés en su aplicación para el marcado digital de textos, *Natural Language Watermarking (*NLW). Entre las múltiples variantes de estas técnicas encontramos algoritmos basados en modificaciones léxicas (Chapman y Davida, 1997), modificaciones sintáctico-semánticas (Muñoz, Argüelles y Carracedo, 2009; Muñoz y Argüelles, 2012), ocultación basada en la traducción entre lenguas (Grothoff *et al*., 2005; Meng *et al*., 2010), ocultación basada en errores ortográficos y tipográficos (Topkara, Topkara y Atalla, 2007) u ocultación basada en la estructura y el formato de un texto (Bergmair, 2007). Con respecto a estas técnicas basadas en la modificación de textos existentes y, al margen de otras cuestiones lingüísticas y estadísticas, existe el problema de mantener en secreto o de destruir el texto original o portador en el cual se realizarán las modificaciones y mediante el cual se creará el estegotexto resultante con la información enmascarada. Es una cuestión importante debido a que si un potencial estego-analista pudiera realizar comparaciones entre el texto original y el estegotexto creado se le simplificaría la tarea de detección. Por estos motivos, el método alternativo basado en la generación automática de estegotextos centra aquí nuestro interés. En primer lugar, el estegotexto generado, que depende de la información que se va a ocultar, puede mejorarse teniendo en cuenta todo tipo de consideraciones lingüísticas y estadísticas en el proceso automático de creación. Por otro lado, este método facilita la creación de un estegotexto único por cada comunicación enmascarada que se quiera realizar, lo que complica el trabajo de un potencial estegoanalista. Esta idea, que podría sonar apasionante, en la práctica resulta de una enorme complejidad, ya que, si bien es viable generar textos con validez léxica y sintáctica, la semántica y la coherencia global son a día de hoy temas sin una solución clara para textos de una longitud media de centenas de palabras y temática variada.

Partiendo de las ventajas que ofrece un sistema frente a otro y conscientes de sus posibles limitaciones, en este trabajo se analiza la posibilidad de generar estegotextos automáticamente en lengua española utilizando modelos N-Gram y se proporciona un concepto novedoso de edición manual de estegotextos a posteriori que facilita la producción de estegotextos de apariencia similar a aquellos escritos por humanos.

En lo que sigue, para facilitar la comprensión del lector, la propuesta se estructura de la siguiente manera: En el apartado 2 se incluye una recopilación de las investigaciones más significativas sobre la generación automática de estegotextos; en el apartado 3 se enumeran nuestras hipótesis y se describe la experimentación con el algoritmo propuesto; finalmente, en el apartado 4 se sintetizan las conclusiones de la investigación realizada y se proponen líneas para posibles trabajos futuros.

2. GENERACIÓN AUTOMÁTICA DE ESTEGOTEXTOS: ESTADO DE LA CUESTIÓN

Los algoritmos de generación automática de textos deben considerar la calidad léxica, sintáctica y semántica, así como la cohesión y la coherencia del

estegotexto resultante. Para aproximarse a este problema, desde finales del siglo XX se han propuesto dos grandes líneas de generación, que se pueden aplicar conjuntamente: unas basadas en imitación gramatical y otras basadas en imitación estadística de un texto "típico" en una lengua concreta.

a) Modelado estadístico del lenguaje natural e imitación estadística de textos de entrenamiento

Para la generación de textos en lenguaje natural se parte de la idea de que las palabras y las expresiones presentes en una lengua siguen patrones concretos. Un modelado estadístico del lenguaje natural permitiría cuantificar diferentes aspectos sobre textos en una lengua concreta que tendría utilidad para la creación de textos con validez lingüística que no sólo no levanten las sospechas de un software automatizado (una máquina) sino tampoco de un lector humano. En general, puede ser muy complejo realizar un modelado estadístico preciso sobre una lengua. Por ello, en determinados entornos, como pueda ser la esteganografía, modelados estadísticos más sencillos serían, en principio, prácticos para propuestas reales. Por ejemplo, analizar la estadística de unos textos de entrenamiento que se toman como referencia. Si el modelo estadístico estuviera basado en textos de entrenamiento, los estegotextos generados basados en ellos reproducirían de una manera u otra su estructura. De hecho, esta idea es recurrente en algunas de las propuestas conceptualmente más interesantes de generación automática de estegotextos basada en imitación estadística.

Un ejemplo significativo es la propuesta de Peter Wayner en 1992 (Wayner, 1992) analizada en su aplicación a la lengua inglesa. La propuesta de Wayner se centra en la generación automática de estegotextos basada en la imitación estadística de uno o más textos fuente (S). La idea conceptual es sencilla: *Cójase una función de imitado f que modifique un fichero A de forma que asuma las propiedades estadísticas de otro fichero B. Es decir, si p(t,A) es la probabilidad de que una cadena t suceda en A, entonces una función de imitado f, hace que la p(t,f(A)) sea aproximadamente p(t,B) para toda cadena t de tamaño menor que n.* La complejidad del modelo estadístico de imitado, el análisis de frecuencia, depende, precisamente del orden estadístico n que es el orden de complejidad del algoritmo. Según esta idea, Wayner definió el siguiente algoritmo de imitado:

1. Construir una tabla con todas las diferentes combinaciones de n letras que ocurran en S y contabilizar el número de veces que ocurren en S.
2. Elegir una de ellas aleatoriamente que actuará de semilla inicial. Esto generará las primeras n letras de T (el estegotexto).
3. Repetir esta acción hasta que se genere todo el texto deseado.

    a. Coger las n-1 letras siguientes de T.

    b. Buscar en la tabla estadística creada todas las combinaciones de letras que comienzan con esas n-1 letras.

    c. La última letra de esas combinaciones forma el conjunto de posibles elecciones para la siguiente letra que será añadida a T.

    d. Elegir entre esas letras y usar la frecuencia de sus ocurrencias en S para "evaluar" cuál es la mejor elección.

    e. Añadirla a T.

Según este algoritmo, un primer orden de imitado genera caracteres aleatorios de acuerdo a su distribución estadística. En un segundo orden se imita la distribución de parejas de caracteres de los textos S de entrenamiento, y así sucesivamente para mayor orden. Se supone, por la información publicada (Wayner, 1992), que en su aplicación a la lengua inglesa, dependiendo del texto y del orden en textos de al menos decenas de KB y orden mayor que 6, pueden obtenerse estegotextos con validez léxica y sintáctica e incluso, algunos, aunque no exento de errores, con apariencia semántica-estructural.

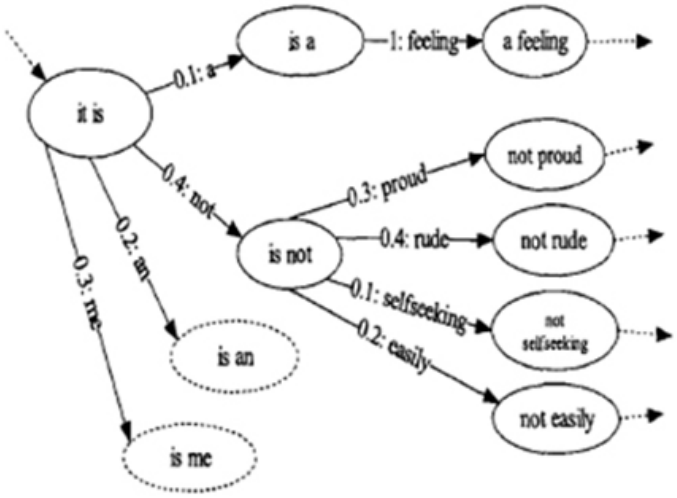
El proceso de ocultación de información se realiza mediante la selección de las opciones de la próxima letra a mostrar. Wayner justificó cómo esto se podría hacer, entre otras opciones, utilizando un árbol de Huffman. El algoritmo esteganográfico, basándose en las frecuencias de aparición de la próxima letra a mostrar, crea un árbol Huffman con ellas, asignándoles un código binario a cada una, este código es el correspondiente a la información binaria necesaria para alcanzar el nodo en el cual está presente la letra en el árbol Huffman. La ocultación de información consistiría en ir eligiendo nodos del árbol, cuyo código–rama coincidan con la información a ocultar. Si la selección de las ramas de este árbol, que imita la estadística de la fuente, es aleatoria, el texto resultante imitará o se aproximará mejor a la distribución estadística del texto fuente. Si imita la estadística es razonable que imite la sintaxis y gramática del texto de entrenamiento.

Conceptualmente no se han propuesto alternativas al sistema de Wayner. Quizás una posible variante esté

ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

basada en cadenas de Markov. Una cadena de Markov puede describirse como un modelo estocástico en el cual la probabilidad de que suceda un evento depende exclusivamente del evento anterior. Esto tiene aplicación en esteganografía lingüística y estegoanálisis. En 2003, Shu-feng investigaría sobre (Shu-feng y Huang, 2003) la posibilidad de generar estegotextos mediante una señal procedente de una fuente de señales basadas en un modelo de Markov. En esta propuesta la sucesión entre elementos, y por tanto los elementos a elegir, se seleccionan con la misma probabilidad. Pero esta investigación no es especialmente útil para nuestros fines, ya que como concluyeron Meng y sus colegas en 2009 (Meng *et al.*, 2009), la elección de los candidatos que suceden a una palabra o expresión deben considerar su estadística de aparición y no ser elegidos como si se produjesen con la misma probabilidad porque, en este caso, los ataques estegoanalíticos se simplifican notablemente. Aunque, por desgracia, el estudio no se realizó con profundidad, en 2009, Dai et al. introdujeron la novedad de generación de estegotextos utilizando cadenas de Markov pero esta vez considerando en todo momento la probabilidad de las palabras a seleccionar en función de las palabras antecesoras y de las que se encontraban en su mismo "nivel/estado" del proceso.

Figura 1. Cadena de Markov con probabilidades entre palabras



b) Modelado gramatical del lenguaje natural y gramáticas libres de contexto

Los textos en lenguaje natural pueden verse como un conjunto de léxico (palabras) que mediante uniones (reglas gramaticales) permiten construir fragmentos con significado (semántica) cuya unión (coherencia) aporta un valor concreto al lector. Dado que de una forma simplista un texto puede verse como un conjunto de oraciones unidas, tiene sentido analizar la posibilidad de imitar la estructura gramatical de una lengua concreta y analizar si en esa imitación para generar un texto válido es posible ocultar información. Una excelente manera de realizar esto es mediante el uso de gramáticas libres de contexto. Para comprender su uso es necesario remontarse a los años 60 del siglo XX. En la década de los 60 el excepcional lingüista A. Noam Chomsky postuló la gramática generativa (1965). Esta gramática se definió como el conjunto de reglas innatas que permite traducir combinaciones de ideas a combinaciones de palabras y en este sentido, *la gramática se convertía en un sistema combinatorio discreto que permite construir infinitas frases a partir de un número finito de elementos* mediante reglas diversas que pueden formalizarse mediante una gramática formal gobernada por normas de transformación. Según esta teoría de lenguaje formal una CFG (*Context-Free Grammar*) se define como una gramática en la que cada regla de producción es de la forma v::=w, donde v es una variable y w es una cadena de símbolos terminales y no terminales. Se entiende por *terminal* la información última de cada regla, por ejemplo, una palabra determinada. Por tanto, en general, una CFG se compondrá de terminales, variables y producciones. Las CFGs han jugado un papel nuclear en el diseño de lenguajes de programación y compiladores, así como en el análisis de la sintaxis del lenguaje natural.

En la década de los 90, Peter Wayner vinculó la posibilidad de utilizar las construcciones CFGs (Wayner,

ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

1995) en la generación de estegotextos de forma automática. Esta idea facilitaría la creación de estegotextos que, al menos, tendrían validez gramatical-sintáctica. El investigador centró sus estudios en su aplicación a la lengua inglesa. A continuación, para facilitar su compresión se añade un ejemplo en lengua española:

Figura 2. Ejemplo de Probabilistic CFG en lengua española en formato BNG. Ocultación máxima de 8 bits.

Variable_Inicio S ::= *AB (.5) | AC (.5)*
A ::= "Buenos días,"(.25) | "Buenas tardes,"(.25) | "Buenas noches," (.25) | "Hola"(.25)
B ::= "estimado amigo" C (.5) | "estimado compañero" C (.5)
C ::= "Juan," D (.25) | "Pedro," D (.25) | "Lucas," D (.25) | "Tomás," D (.25)
D ::= "quedamos algún día para" E (.5) | "dame tu número de teléfono para" E (.5)
E ::= "hablar" F (.5) | "charlar" F (.5)
F ::= Un saludo (1.0).

La ocultación de información se realiza mediante la selección de elementos concretos dentro de una regla específica, regla que es elegida mediante algún algoritmo de selección concreto. En el ejemplo anterior, una posible oración extraída (selección de la regla AB) de las reglas definidas podría ser: "Buenos días, estimado compañero Tomás, dame tu número de teléfono para charlar. Un saludo" la cual ocultaría 8 bits (1+2+1+2+1+1).

Wayner desarrolló varios ejemplos interesantes aplicando estas ideas: spammimic (ocultación en un mensaje con estructura de correo de spam), baseball game, etc. Aunque Wayner se esforzó en formalizar la construcción de CFGs seguras con utilidad esteganográfica (Wayner 1995), es cierto que su utilidad esteganográfica debe ser muy matizada. El primer problema es que la gramática debe permanecer privada, emisor y receptor la tienen que conocer, ya que si no es así un atacante podría inferir fácilmente la información oculta. Este problema es mayor si la gramática es estática-manual. El segundo problema es que si esta gramática fuera conocida por el atacante forzaría al emisor y al receptor a un nuevo proceso tedioso (manual) y costoso de generación de una nueva gramática. El tercer problema es que la calidad del estegotexto depende claramente de la gramática y si esta tiene pocas reglas es más que probable la repetición de frases y términos en el estegotexto, facilitando a los estegoanalistas su trabajo. Además, aunque la gramática sea generada automáticamente de uno o más textos de referencia, conocidos por emisor y receptor, deben considerarse otros análisis al generar algoritmos esteganográficos basados en CFGs, por ejemplo, las palabras (términos) en una CFG se relacionan con sus vecinos en formas fijas. Aunque se añadan modelos estadísticos a las gramáticas como son las Probabilistic Context Free Grammars, PCFG, para dificultar ataques de análisis, siempre existirán correlaciones mutuas si se quiere que el texto sea coherente para un humano. Por último, deben considerarse los ataques basados en estudio de terminales, información última de cada regla, ya que aunque las variaciones de los textos creados puedan crecer sustancialmente con el tamaño de una gramática dada, el número de terminales está limitado por el tamaño de la gramática, lo cual significa que forzosamente, si el texto es lo suficientemente grande, se tienen que producir, y por tanto repetir, combinaciones lineales de terminales.

En la práctica, resulta realmente complejo utilizar CFGs en herramientas públicas de manera robusta en la concepción actual. Un intento relevante de los pocos destacables, fue el sistema NICETEXT, del que se pueden extraer ideas para nuevos diseños.

En 1997, Chapman y Davida en diversas investigaciones (Chapman y Davida, 1997; Chapman, Davida y Rennhard, 2001) desarrollaron un sistema software, NICETEXT, que permite generar modelos gramaticales basados en la posibilidad de imitar la gramática de uno o varios textos de entrenamiento. Estas reglas gramaticales, a modo de elementos etiquetados de una oración, permiten la generación de frases del estegotexto resultante.

NICETEXT permite la generación dinámica de las reglas gramaticales basada en la imitación gramati-

cal de textos de entrenamiento, es decir, habilita los procedimientos necesarios para identificar reglas sintácticas y mediante un etiquetador PoS (*Part of Speech*), *pckimmo*, permite definir qué tipo de palabra (categoría lingüística) forma cada elemento de la regla sintáctica generada (verbo, nombre, adjetivo, etc.). La novedad de esta herramienta reside en la forma de ocultación y en la recuperación de la información ocultada, ya que el receptor no necesita conocer la gramática utilizada por el emisor. La herramienta está basada en la sustitución de los elementos etiquetados en cada regla por palabras categorizadas por contenido semántico. Este esquema tiene dos ventajas importantes: La primera, dado que la información se oculta independientemente de la gramática, el emisor, en el peor de los casos, podría utilizar una gramática única por comunicación y de la riqueza que desee; la segunda, que la ocultación de información se realiza mediante la selección de una palabra dentro de una categoría de un diccionario categorizado, este diccionario es compartido por el emisor y el receptor y debe permanecer secreto.

Adicionalmente a las características mencionadas, es importante añadir que NICETEXT todavía produce estegotextos con defectos derivados de sustituciones no válidas en contexto y anomalías entre el estilo de escritura seleccionado y el vocabulario empleado. No obstante, pocas innovaciones se han dado en esta línea de investigación desde las ventajas conceptuales aportadas por NICETEXT, aunque es cierto que se han publicado otras propuestas basadas en gramáticas libres de contexto, como por ejemplo la herramienta TEXTO que transforma información a sentencias en inglés (disponible en ftp://ftp.funet.fi/pub/crypt/steganography), las herramientas C2txt2c (Zuxu *et al*., 2007) y Csteg (Blasco *et al*., 2008) que ocultan código fuente de lenguajes de programación en oraciones en lenguaje natural o el sistema Lunabel (Chand y Orgun, 2006).

3. PROPUESTA DE ALGORITMO DE GENERACIÓN AUTOMÁTICA DE ESTEGOTEXTOS BASADO EN MODELO N-GRAM Y LEY DE ZIPF

El estudio de las propuestas publicadas de generación automática de estegotextos hace que centremos nuestro interés en la posibilidad de generación basada en imitación estadística. El principal motivo reside en evitar posibles restricciones o limitaciones del uso de todos los recursos necesarios para llevar a la práctica una generación basada en imitación gramatical: analizadores morfosintácticos, recursos léxicos, etiquetadores, etc. Al fin y al cabo, si la imitación estadística es la adecuada, se debería imitar la estructura gramatical de la lengua o textos de entrenamiento seleccionados.

En el siguiente apartado se proponen una serie de hipótesis acerca de la posibilidad de utilizar modelos N-Gram para generar automáticamente estegotextos en lengua española. Seguidamente, se llevan a cabo diversas medidas para corroborar las afirmaciones realizadas.

3.1. Hipótesis lingüísticas, estadísticas y definición del algoritmo

Nuestra investigación se centra en la posibilidad de adaptar el algoritmo de Wayner para la imitación estadística de la relación entre palabras en un texto de entrenamiento dado. Como adelantábamos, la propuesta de Wayner hacía hincapié en la posibilidad de imitar estadísticamente la aparición de caracteres en un texto en lengua inglesa. Esta imitación, considerando secuencias de 6 o más caracteres consecutivos, permitía generar estegotextos con validez léxica y sintáctica, e incluso, en ocasiones, con apariencia de texto con coherencia global. Esta propuesta, no exenta de pequeños fallos, producía, debido a su diseño, errores léxicos. Las palabras se troceaban en un número determinado de caracteres y era posible, por ejemplo, al seleccionar el primer conjunto de letras de la tabla raíz que al juntarse no produjeran palabras válidas. Ante esta situación se plantean las siguientes hipótesis:

Hipótesis 1. Parece más lógico imitar textos de entrenamiento a nivel de palabra en lugar de a nivel de carácter. Así, se propone que la palabra sea la unidad mínima en la que se divida un texto a imitar, y de este modo se eviten de raíz las inconsistencias léxicas.

Hipótesis 2. Un modelo estadístico N-Gram permitiría generar automáticamente estegotextos de calidad; un modelo estadístico N-gram que permita contabilizar las ocurrencias y la posición de posibles palabras en una secuencia de n palabras consecutivas.

Dado uno o más textos de entrenamiento, sería posible anotar las ocurrencias de las palabras presentes en los mismos y contabilizar la repetición de cada una, es decir, anotar qué palabras van detrás de qué otra y con qué probabilidad. Un algoritmo de generación automática de estegotextos podría aprovecharse de esa información para imitar la estructura de un texto en lenguaje natural seleccionando palabras. De hecho, si la selección de las palabras fuera aleatoria

ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

sería más probable que las palabras más probables se seleccionaran y menos el resto, es decir, se imitaría la estadística de la fuente de entrenamiento. Si se imita la estadística de la fuente, es más probable que el estegotexto resultante, que está basado en textos de entrenamiento con validez léxica y sintáctica, tenga a su vez validez léxica y sintáctica. La ocultación de información se realizaría mediante la selección concreta de unas palabras u otras. Los textos resultantes deberían tener la suficiente calidad lingüística para no ser detectados por un software automatizado y en la medida de lo posible tampoco por lectores humanos.

Hipótesis 3. De la experiencia probando el algoritmo de Wayner se consideran los signos de puntuación como palabras individuales: **. , ; : ¡ ? ,)] ¡ ¿ ([**. La justificación de que esto sea así se debe principalmente a dos razones: la primera, está relacionada con la posibilidad de utilizar técnicas correctoras en el texto resultante (esto se verá en próximos apartados) y con el tamaño del estegotexto final. La segunda, si se separa una palabra de un signo de puntuación determinado será más probable que existan más palabras que le sucedan que si esto no se hubiese hecho: a más sucesores será posible ocultar más información generando menos palabras. Por ejemplo, es más probable que la palabra "mi" tenga varias sucesoras que la palabra "¿mi". Estas dos mejoras facilitarán obtener estegotextos con mayor validez léxica y sintáctica. El imitado a nivel de palabra facilitará a su vez el uso de técnicas correctoras, como se verá posteriormente, mediante la edición manual, para generar estegotextos de una elevada calidad lingüística.

Considerando estas hipótesis se formula la implementación del siguiente algoritmo:

1. El proceso de generación se basa en el análisis de bloques de n palabras, extraídas de uno o más textos de entrenamiento, mediante una ventana deslizante que se desplaza una posición para cada nuevo bloque. Es decir, el primer bloque tendrá las palabras de 0 a n-1, el segundo bloque de 1 a n, y así sucesivamente.
2. N define el orden de complejidad del algoritmo (N-gram), lo que significa el número de palabras a considerar consecutivamente.
3. Las palabras se relacionan mediante nodos enlazados en los que se contabiliza el número de veces que se han repetido en el texto de entrenamiento. Según esto, existirá una tabla raíz que almacenará todas las "palabras diferentes" que existan en el texto fuente.

Basado en lo anterior, el algoritmo de generación de estegotextos funcionaría, en general, de la siguiente manera:

a) Se selecciona una "palabra" aleatoriamente de la tabla raíz, aunque podría considerarse otro criterio con fines sintácticos, como por ejemplo hacer que el estegotexto empezase por un artículo o por mayúscula. Esta selección desencadenará el resto del estegotexto resultante, por lo que para un mismo texto de entrenamiento es posible obtener diferentes estegotextos seleccionando diferentes palabras de la tabla raíz.

b) Si esta palabra no tiene sucesores, es decir, no apunta a otro nodo, se elije otro término de la tabla raíz (paso a). Si el nodo sucesor solo tiene una palabra, esta palabra se añade al estegotexto, lo que significa que no es posible ocultar información en este paso, y se elige el siguiente nodo disponible. Si el nodo sucesor tiene varias palabras posibles entre las que elegir esta selección permitirá ocultar información. Para intentar que la imitación estadística sea lo más adecuada posible, lo que afecta a la calidad lingüística del estegotexto, se decide que con las palabras de cada nodo y las frecuencias con las que aparecen (después de la palabra del nodo anterior) se construya un árbol de Huffman. La palabra seleccionada será aquella cuyo código binario (información binaria para acceder a ella a través de las ramas del árbol) coincida con la información binaria que se quiere ocultar.

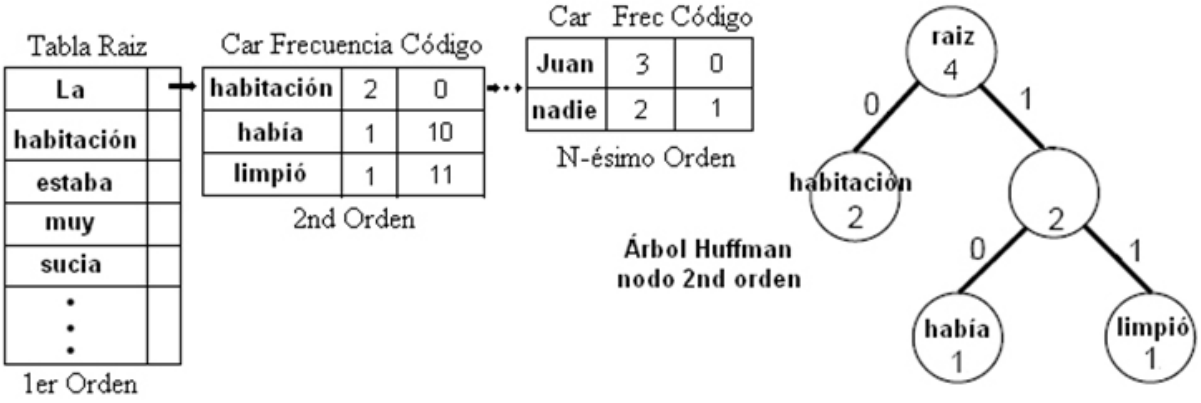
c) Si se llega al último nodo (si el orden n=8 serían 8 palabras consecutivas) se elige la última palabra seleccionada para el estegotexto y se vuelve al paso b). Este proceso se repite hasta que se genere el estegotexto que oculta la información deseada.

d) El receptor necesita construir la tabla de frecuencias del texto de entrenamiento seleccionado e invertir el proceso para conocer los bits que ocultan cada palabra del estegotexto recibido.

Por ejemplo, el algoritmo principal podría elegir de la tabla raíz la palabra "La" seguidamente la palabra "habitación" y finalmente, el orden n-ésimo, la palabra "Juan". De esta forma se construiría el estegotexto con la información a ocultar. El receptor reconstruiría la tabla de frecuencias utilizando los textos de entrenamiento y recuperaría la información enmascarada.

ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

Figura 3. Ejemplo de implementación del algoritmo propuesto en Stelin



3.2. Medidas y experimentación

Con el objetivo de cuantificar la calidad de la propuesta algorítmica de generación automática de estegotextos basada en modelo N-Gram se implementa este algoritmo en lenguaje JAVA y se realizan una serie de medidas, estudiando propiedades estadísticas, el tamaño y la calidad del estegotexto resultante.

En primer lugar es interesante comprobar si un modelo N-Gram imita, se aproxima, correctamente a la estadística de textos de entrenamiento. Si esta imitación es adecuada los estegotextos resultantes imitarán de mejor manera la estructura gramatical de los textos de entrenamiento. Analizando diferentes fuentes/textos de entrenamiento puede cuantificarse como es cierta esta aproximación estadística. Al menos lo es comparando la estadística de caracteres y de palabras en fuentes de entrenamiento y estegotextos generados para órdenes n mayores de 7.

A continuación, por brevedad, se adjuntan algunos ejemplos de estas medidas, véase Figura 4, 5 y 6.

Figura 4. Ejemplo de comparación de la distribución de frecuencias de caracteres de una fuente de texto de entrenamiento (los 13 primeros capítulos del Quijote con un total de 29.805 Palabras) y un estegotexto generado de ocultar 256 octetos (2048 bits) de información oculta a partir de dicha fuente (como nivel de atomicidad la palabra)

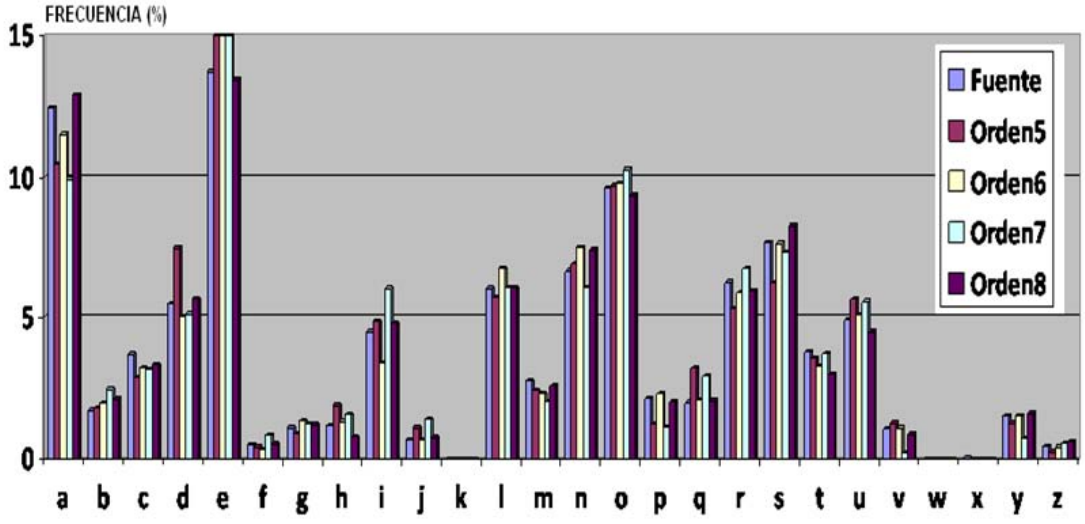


Figura 5. Comparación de frecuencias de las palabras más probables en un estegotexto que oculta 16 octetos (128 bits) mediante un orden 10. Texto Fuente: RIMAS (42 KB) de Gustavo Adolfo Bécquer. Factor expansión 1:144

Palabra	Fuente – Estego	Palabra	Fuente – Estego	Palabra	Fuente - Estego
Y	3,52% - 4,12%	De	3,91% - 2,98%	A	1,59% - 1,37%
En	3,61% - 4,12%	Las	1,4% - 1,83%	Yo	1,21% - 1,14%
El	3,87% - 3,89%	Se	1,10% - 1,6%	Con	0,74% - 1,14%
Que	3,90% - 3,66%	Un	1,33% - 1,37%	Me	0,74% - 0,91%
La	3,58% - 3,66%	Mi	0,6% - 1,37%		

Figura 6. Ejemplo de comparación de frecuencias de las palabras más probables en un estegotexto que oculta 256 (2048 bits) octetos mediante un orden 10. Texto Fuente: RIMAS (42 KB) de Gustavo Adolfo Bécquer. Factor expansión 1:166. Tamaño de estegotexto generado comparable al tamaño de la fuente

Palabra	Fuente – Estego	Palabra	Fuente – Estego	Palabra	Fuente - Estego
Que	3,9% - 5,6%	De	3,91% - 3,44%	se	1,10% - 2,20%
La	3,58% - 4,39%	Del	1,10% - 3,13%	Los	1,25% - 1,6%
En	3,61% - 4,36%	Yo	1,21% - 3,11%	Al	1,19% -1,56%
Y	3,52% - 3,87%	El	3,87% - 3,05%	Par	0,04% - 1,49%

El algoritmo desarrollado produce estegotextos cuyo tamaño depende del orden n y de los textos de entrenamiento (factor de expansión). Cuanto mayor sea el número de palabras presentes en un texto de entrenamiento más probable será que existan múltiples alternativas de palabras a continuación de otra dada, por tanto, habrá más opciones entre las que elegir y se conseguirá ocultar más información por palabra generada, lo cual reducirá el tamaño final del estegotexto resultante. Por otro lado, el orden de imitado también afectará directamente al tamaño final. Si el orden es pequeño, por ejemplo n=3 (tres palabras consecutivas), es más fácil que existan varias palabras sucesoras en el texto de entrenamiento que si el orden es mayor. Esto es fácil de entender con un ejemplo: es más fácil encontrar varias palabras que sucedan a "la casa", por ejemplo "la casa es", "la casa blanca", "la casa está", etc., que palabras que sucedan a "yo vivo en la calle Carretas situada en". Un orden pequeño generará estegotextos más pequeños pero de peor calidad lingüística, mientras que un orden grande generará estegotextos más grandes pero de mejor calidad. Por tanto, la elección del orden es una cuestión arbitraria que debe considerar el emisor en función del texto de entrenamiento.

ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

En la práctica, las pruebas realizadas indican, que en general un orden entre 7 y 9 proporciona unos resultados léxicos y sintácticos que no son mejorados con un orden superior. En la medida de lo posible, debe evitarse el uso de órdenes mayores de 9 ya que provocaría un estegotexto de mayor tamaño. El impacto final de los elementos en juego (orden n, texto de entrenamiento e información a ocultar) influirán en el factor de expansión. Se conoce como factor de expansión a la relación:

$$Factor\ Expansión = \frac{Tamaño\ del\ estegotexto\ resultante}{Tamaño\ del\ mensaje\ secreto}$$

En la Figura 7 y a modo aclaratorio se puede observar el crecimiento cuasi-lineal que experimenta el tamaño del estegotexto resultante para diferentes órdenes y diferentes volúmenes de información a ocultar para un texto de referencia de concreto. Por ejemplo, si se desea ocultar 32 bytes (256 bits) con un orden n=9 el factor de expansión será de 95, es decir, el resultado final será 32*95 = 3040 bytes (1:95).

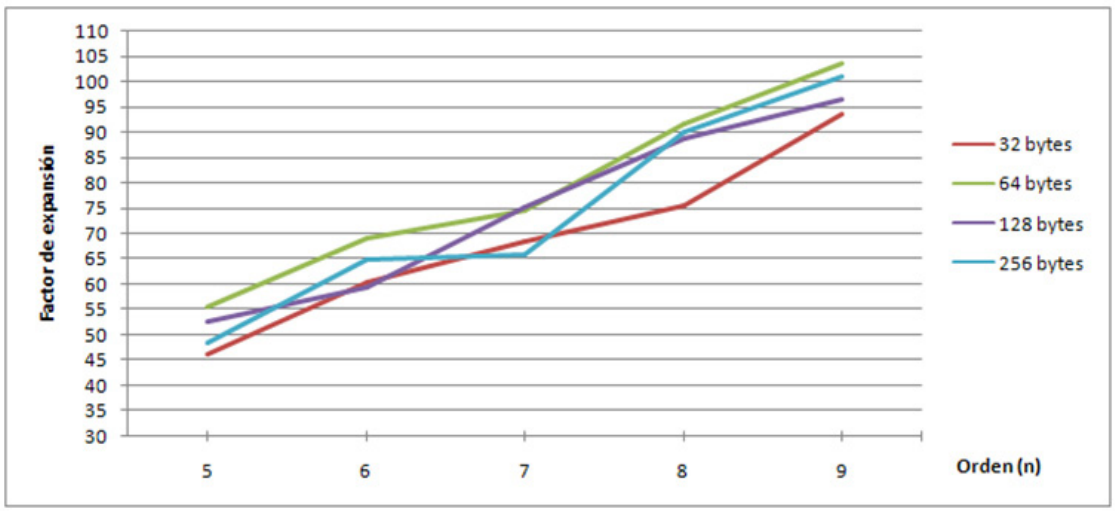
Por tanto, en la cuantificación de que texto de referencia será mejor para reducir el factor de expansión debe considerarse:

- El orden de entrenamiento. A menor orden menor factor de expansión por la explicación reflejada en párrafos anteriores.
- La probabilidad de las palabras en el texto de referencia. Cuanto mayor sea el número de palabras con probabilidad de ocurrencia alta en un lenguaje dado existirá un mayor número de palabras a continuación de las mismas y por tanto se podrá ocultar más cantidad de información reduciendo el factor de expansión (al que le afecta en gran medida las palabras seleccionadas que no ocultan información).

Una vez observados aspectos estadísticos y de tamaño derivados de la aplicación de un modelo N-Gram en la generación de estegotextos es el momento de analizar la calidad de los textos generados.

A continuación se presentan varios ejemplos de estegotextos generados para diferentes órdenes y textos de entrenamiento, con el objetivo de apreciar algunas de las afirmaciones vertidas en los párrafos anteriores (Figura 8 y Figura 9). Los estegotextos que se muestran a continuación poseen diversas imperfecciones que levantarían la sospecha de un lector humano aunque estadísticamente fueran "parecidos" a un texto escrito por un humano. Estos ejemplos se muestran sin aplicar ningún tipo de procedimiento de mejora a los mismos para observar ejemplos "en bruto" del resultado de un algoritmo de imitado N-gram. Como se puede observar, las inconsistencias son más detectables cuando el estegotexto es de mayor longitud.

Figura 7. Relación orden de complejidad y factor de expansión (mensaje a ocultar-estegotexto creado) para diferentes mensajes a ocultar de pequeño tamaño. Texto original: versión digital de Poesías Completas (290KB) de Antonio Machado (51.531 palabras)



ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

Figura 8. Ejemplo de estegotexto generado automáticamente en lengua española. Ocultación 132 bits. Texto fuente versión digital de "100 años de Soledad" de Gabriel García Márquez. 0,9230 bits/palabra estegotexto. Tamaño fuente: 809KB texto plano, 137.649 palabras. Orden de complejidad 9. Expansión 1:43

> acostumbraba a su estado de vigilia, empezaban a Úrsula, fue el daguerrotipo de Remedios en Macondo. Tan pronto como José Arcadio cerró los ojos. Entonces el padre Nicanor se le había apagado el rescoldo del corazón. La primera vez que estuvo en Manaure después estaba de regreso con seis soldados descalzos y le dio recetas de bebedizos que en casos de percances hacían expulsar «hasta los remordimientos de la ciénaga, y los gitanos confesaron que el tiempo había hecho en la casa, como si el destino hubiera invertido la situación política tan confusa que cuando ordenó restaurar la mano de él estaba sudorosa y helada. Era un hombre cambiado. Los ciento veinte centavos», dijo con voz de-solada. La primera noticia directa que Úrsula recibió de él, y tuvo que buscarse otro entretenimiento para que la destriparan los niños En cambio cuando se supo

Figura 9. Ejemplo de estegotexto generado automáticamente en lengua española. Ocultación 132 bits. Texto fuente versión digital de "Poesías Completas" de Antonio Machado. 0,7292 bits/palabra estegotexto. Tamaño fuente: 290KB texto plano, 51.461 palabras. Orden de complejidad 9. Expansión 1:45

> adheridos a una emoción humana. El poeta, en el vacío insensible; y aunque es ya pensado como aquello que absolutamente no es, por cercano al sujeto consciente, más que un hombre al uso que sabe su obra terminada, "Ya estoy en el secreto. Bástele a usted, por ahora, en la sombra, fiad en mi espada... Mi espada se ha visto a la luna, orilla al mar salado, y el sol de fuego. A un paso de la mar cual la nube y la tormenta; es el Criador y la criatura lo ha querido— en la mano creadora del olvido... Mientras traza su curva el pez de fuego borbollar. Sentado ante una mesa de un arquero en torno a Soria. —Soria es una barbacana, hacia Aragón, que me ahoga fluye en esperanza de Ella... En su claro verso se canta y medita sin color, desubstanciado y frío, lleno de una tarde inmensa; mas falta el campo, ya se ilumina; allí un día, en la ancha mar violeta hunde el sueño su pétrea escalinata, y hace temblar el postigo, y suena en los

En los ejemplos anteriores, en los que, de momento, ignoramos inconsistencias y problemas de cohesión y coherencia global, puede observarse que los estegotextos generados no finalizan necesariamente con una estructura puramente sintáctica. Esto no es un problema excesivo ya que se pueden finalizar manualmente o mediante algún procedimiento automático. El algoritmo puede insertar en la información ocultada un marcador de fin de forma que no implique problemas de desincronización en el receptor al recuperar la información. El receptor sabe hasta qué palabra el estegotexto contiene información útil y después de cuál es simplemente relleno.

La calidad del texto de entrenamiento, como lo es también el orden de complejidad, es vital en la generación de los estegotextos. Entendemos por calidad no sólo el tamaño del mismo, sino también su validez léxica, sintáctica, su género literario, su estilo, etc. Ante este condicionante una pregunta interesante a resolver sería qué texto de entrenamiento, si es que hay alguna preferencia, es más adecuado para su uso esteganográfico. En principio, diferentes tipos de textos podrían ser considerados como fuente para ocultar información: poemas, novelas, artículos periodísticos, código de programación, etc. Desde un punto de vista lingüístico, y al trabajar con secuencias de n palabras consecutivas, sería interesante filtrar zonas del texto que pudieran afectar claramente a la coherencia del estegotexto resultante como resultado de concatenar diversas secuencias de palabras. Por ejemplo, descartar índices, títulos, numeraciones (a), b), c), I, II, III), fechas, referencias, etc. En la práctica los textos de entrenamiento recomendados, según las investigaciones actuales, serían libros y novelas (narrativa) con decenas de miles de palabras. También una obra de poesía podría ser una buena candidata como fuente de entrenamiento ya que, en general, la coherencia entre las frases en este tipo de textos puede entenderse como más flexible que en otros tipos de textos de entrenamiento. De hecho, es frecuente en este tipo de textos que frases adyacentes hablen de cosas muy diferentes.

ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

Como se justificó en apartados anteriores, el algoritmo permite la creación de diferentes estegotextos para una misma información a ocultar, texto de entrenamiento y orden n. Esto depende de la primera palabra seleccionada en la tabla raíz. Será posible crear tantos estegotextos diferentes como palabras diferentes existan en el texto de entrenamiento y muchos de estos estegotextos tendrán similitudes.

Mediante el software implementado es posible generar todos los estegotextos posibles, es decir, generar cada uno de los estegotextos para cada una de las palabras de la tabla raíz. Esta opción facilitará al emisor elegir el estegotexto más adecuado, más bits ocultos por palabra (estegotextos más pequeños) o aquellos con mejor aceptabilidad lingüística. Esta opción es muy interesante si se utiliza un texto de entrenamiento diferente por cada comunicación y se desea elegir el "mejor" estegotexto posible.

A continuación, se adjunta un ejemplo de algunos de los posibles estegotextos para la ocultación de una pequeña información de 72 bits, que contiene la dirección IP 81.10.49.51 representando cada carácter a ocultar con 6 bits y después 6 bits de código de fin, utilizando para ello como fuente de entrenamiento la versión digital de "Guerra y Paz" de León Tolstoi, cuyo tamaño en texto plano es 396KB con 69.344 palabras. En los siguientes ejemplos se ha utilizado un orden n=7, generando 9.291 estegotextos diferentes.

-_STEGOTEXT_-153

nueva ciencia - la estrategia -, pero el príncipe Bagration no le miró con gozo, inclinando la mano, cosa que todavía no le ha llamado ni una sola en el campo. -No olvides que estarás con mi padre y la hija de la visitante, con la túnica de las tres días de armisticio. Bagration contestó. - ¿No quieres...? - preguntó. - A la habitación de

-_STEGOTEXT_-160

había hecho un esfuerzo extraordinario para decir que no respondería de sí misma frase -. No se pueden vivir sin guerras. ¿Por la noche, cuando, después de la ausencia, a las piernas. Sus mejillas se contraían violentamente, y cuando se encogió de hombros y abrió los brazos separados del cuerpo, presentaba su marido con el mismo tono de la Princesa, como si ésta se fue a dormir a casa del conde Bezukhov. III En el momento

3.3. Calidad de los estegotextos generados, maquillaje manual y la ley de Zipf

El mayor problema en la ocultación de información mediante esteganografía lingüística, ya sea en textos existentes o generados, es conseguir que el estegotexto resultante presente cohesión y coherencia global, es decir, que no sea una secuencia de frases "más o menos ordenadas".

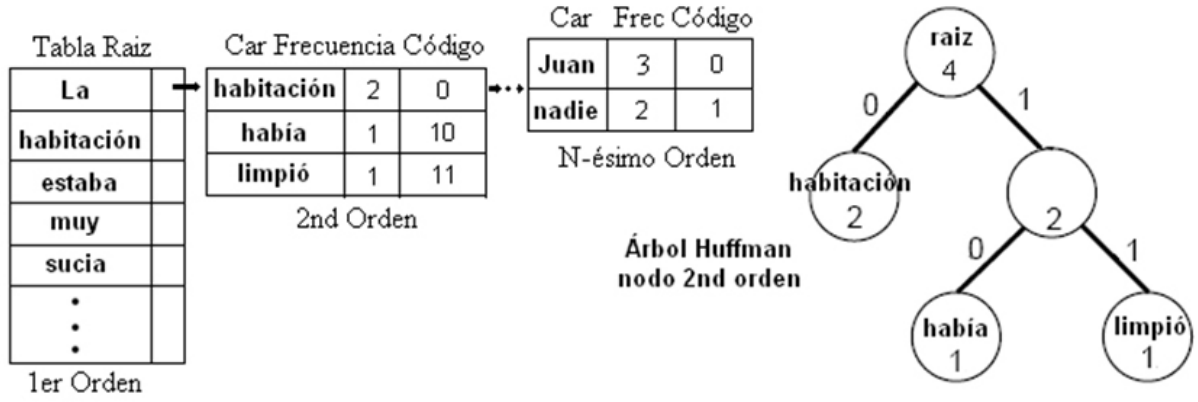
El algoritmo de generación automática basado en un modelo N-Gram produce todavía pequeños errores gramaticales, por ejemplo, signos de puntuación que se abren y no se cierran, así como posibles problemas de coherencia global cuando el estegotexto crece en tamaño, por ejemplo, que se repitan expresiones. Teóricamente es difícil razonar, con los conocimientos actuales, que un algoritmo pueda generar estegotextos de muy alta precisión para contextos variados, ya que esto sería como haber encontrado una solución a un problema parecido al formulado en el test de Turing[2]. No obstante, mientras se consiguen mejoras parciales puede introducirse un nuevo concepto que afectará en gran medida a la calidad final del estegotexto resultante: el maquillaje manual del estegotexto a posteriori.

Dado que resulta muy complejo dar con un algoritmo genérico que genere automáticamente estegotextos con una validez lingüística alta, esto es más crítico para estegotextos más grandes, el problema se puede enfocar de otra manera. Si no es posible generar estegotextos perfectos, a lo mejor sí es posible modificarlos una vez creados "maquillando" los errores derivados de la generación automática. Si se profundiza en esta idea, existe el problema fundamental de sincronizar las "palabras" donde el emisor oculta la información con las "palabras" que espera el receptor para recuperar el mensaje oculto, es decir, introducir cambios en el estegotexto generado implicaría que el receptor debería conocerlos, lo cual no es muy práctico porque implicaría el envío de los mismos para evitar la desincronización.

Pensemos en la estructura en forma de árbol utilizada por el algoritmo para ocultar una información. Una información se oculta mediante la selección de una palabra entre las disponibles en el siguiente nivel. Es decir, si se selecciona la palabra "La" la siguiente palabra podría ser "habitación" (bit 0), "había" (bit 10) o "limpió" (bit 11) en función de los bits a ocultar, véase Figura 10. El receptor al recibir el estegotexto construiría la tabla de frecuencias al igual que el emisor e iría relacionando palabra por palabra el estegotexto recibido con la tabla generada, de esta forma, el receptor al recibir "La habitación", "La había" o "La limpió" sabría que se ha ocultado un bit (0) o dos (10-11).

ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

Figura 10. Ejemplo de codificación de palabras y árbol Huffman



En este proceso de recuperación y de sincronización podría actuarse de varias formas: Una consistiría en que la herramienta diera error si después de una palabra del "supuesto" estegotexto se encuentra otra palabra que no coincide con ninguna de las esperadas. Si el atacante tuviera información de las tablas de frecuencia esto le simplificaría descartar mensajes sin información oculta; otra variante consistiría en despreciar todas las palabras que se lean hasta que se encuentre una de las palabras posibles para el nivel de la tabla (lista enlazada) donde nos encontremos. Realmente lo importante es que el receptor no pierda la sincronización respecto a la tabla de frecuencias y al estegotexto recibido.

Este detalle, permite mejorar a posteriori, en principio manualmente, estegotextos generados por el emisor sin que ello afecte al receptor. La única condición es que el emisor puede utilizar cualquier palabra que no se encuentre en el nivel posterior para que el receptor no pierda la sincronía. Es decir, en el ejemplo anterior entre las palabras que forman las parejas "La habitación", "La había", "La limpió", podría utilizarse cualquier palabra, una o más, que no fuera "habitación, había o limpió". De esta forma el emisor puede corregir posibles errores gramaticales y mejorar la coherencia del texto sin necesidad que el receptor conozca esta información.

Por ejemplo, si ocultamos una pequeña información de 126 bits, usando un orden 9 y fuente de entrenamiento las poesías completas de Antonio Machado obtendríamos entre los estegotextos posibles uno como el siguiente (se adjunta sólo un fragmento, Figura 11):

Figura 11. Estegotexto que oculta 126 bits. Texto fuente versión digital de "Poesías Completas" de Antonio Machado. 0,7636 bits/palabra estegotexto. Tamaño fuente: 290KB texto plano, 51.461 palabras. Orden de complejidad 9. Expansión 1:48

planeta por donde cruza errante la sombra de Caín criminal. ¡Gloria a Caín hoy sólo quedan lágrimas para llorar. No hay camino, sino estelas en la mar. ¡Fugitiva ilusión de ojos guerreros, que el polvo barre y la ceniza avienta. ¿Qué has hecho la muerte no hay camino, se hace camino al andar. El que espera desespera, dice la mano viril que la blandiera, no por los salones de sal-si-puedes suena el rebato de la tarde en la arboleda! Mientras el corazón pesado […]

Supongamos por un momento que queremos corregir una serie de pequeños errores, como, por ejemplo, signos de puntuación o mayúsculas (Figura 12).

Figura 12. Algunos de los errores gramaticales presentes en el estegotexto de la Figura 11

> planeta por donde cruza errante la sombra de Caín criminal. ¡Gloria a Caín! Hoy sólo quedan lágrimas para llorar. No hay camino, sino estelas en la mar. ¡Fugitiva ilusión de ojos guerreros, que el polvo barre y la ceniza avienta. ¿Qué has hecho? La muerte no hay camino, se hace camino al andar. El que espera desespera, dice la mano viril que la blandiera, no por los salones de sal-si-puedes suena el rebato de la tarde en la arboleda! Mientras el corazón pesado […]

Para solucionar estos problemas, el algoritmo implementado genera una plantilla con las palabras posibles en cada nivel, de forma que el emisor pueda seleccionar qué palabras añadir entre palabras del estegotexto, palabras que serán despreciadas por el receptor. Por ejemplo, seleccionamos de Figura 11 la frase "*de la tarde en la arboleda! Mientras el corazón*". Veamos un trozo de la plantilla generada:

> [WORD:en] [muerta][flota][.][bella][roja][en][,][sobre][arrebolada][y]
>
> [WORD:la][sus][la]
>
> [WORD:arboleda] [arboleda]
>
> [WORD:!] [!]
>
> [WORD:Mientras] [Mientras]
>
> [WORD:el] [el]
>
> [WORD:corazón][querido][sueño][fondo][mar][temblor][semblante][tictac][vino][aire][sol][ataúd][silencio][blanquecino][maestro][solitario][blanco][mármol][fruto][encanto][hálito][patio][pretil][ambiente]…

Teniendo en cuenta esto, editamos la frase. Entre las múltiples opciones posibles elegimos la siguiente: *"tarde en la* ***dulce*** *arboleda****, ¡qué sensación****! Mientras el corazón".* De esta forma tan sencilla puede mejorarse sustancialmente la calidad del estegotexto generado. Para solucionar problemas derivados de signos de puntuación que se abren y no se cierran (o viceversa) u otros. Como se destacaba anteriormente el hecho de considerar los signos de puntuación como palabras individuales permitirá con esta técnica correctora compensar posibles errores delante o detrás de ellos.

Por suerte, en los textos en lenguaje natural en español, y en otras lenguas, las palabras tienen co-ocurrencias determinadas, es decir, es más probable que ciertas palabras vayan detrás de otras y es más probable que existan más palabras detrás de unas que de otras. Esto tiene que ver con la Ley de Zipf que afirma que un pequeño número de palabras son utilizadas con mucha frecuencia, mientras que ocurre que un gran número de palabras son poco empleadas. En general, existirían pocas palabras después de las cuales será muy costoso (en tiempo) elegir una palabra nueva, porque aparecerán muchas opciones en el nivel de la tabla correspondiente, y muchas palabras después de las cuales existirán pocas opciones, con lo que se tendrá más libertad para añadir palabras nuevas.

Por ejemplo, en textos en lenguaje natural palabras como *de, la, que, el, en, y, a, los*, entre otras, son más probables, luego es más probable que existan más palabras que puedan aparecer con estas. Si nos fijamos en el ejemplo anterior *("tarde en la* ***dulce*** *arboleda,* ***¡qué sensación!*** *Mientras el corazón)* resultaría trivial añadir información antes del artículo "el" (sirve cualquier palabra distinta del artículo *el*) pero sería más difícil encontrar palabras, diferentes a las reflejadas en la plantilla, después de este artículo y antes de la palabra "corazón".

Estos criterios ayudan a reducir el tiempo de edición y las posiciones donde es mejor trabajar para corregir los posibles fallos gramaticales. En resumidas cuentas, es posible generar estegotextos en lengua española automáticamente y corregir los errores manualmente sin afectar a la correcta recepción, creando estegotextos de una calidad notable. Esta aportación original es la que hemos denominado *maquillaje manual* del estegotexto.

No es sencillo estimar la capacidad de ocultación real del algoritmo ya que eso depende del texto de entrenamiento, del orden y de las palabras introducidas mediante edición manual. Las pruebas que se han realizado reflejan tamaños de 0,5 a 2 bits por cada palabra presente en el estegotexto final. Las pruebas realizadas indican que para ocultación de informaciones por encima de la centena de bits, los estegotextos generados son de un tamaño destacable del orden de decenas de centenas de palabras (depende del tamaño y de la "calidad" de la fuente de entrenamiento) y por tanto la edición manual llevará un tiempo considerable (decenas de minutos). El interés de invertir más o menos tiempo en la calidad de los estegotextos generados dependerá de la importancia de la información intercambiada entre emisor y receptor.

Adicionalmente a lo anterior, existe una variante posible del funcionamiento del algoritmo aprovechándose de la técnica correctora publicada. Esta variante consistiría en la generación de estegotextos de "mala calidad" pero que ocuparan poco espacio para dedicar más tiempo a la corrección de pocos errores. Este supuesto puede ser conseguido mediante la utilización de un orden n bajo. Si el estegotexto es pequeño, sería factible incluso, mediante la técnica de maquillaje manual publicada, retocar la coherencia global de todo el texto.

Por ejemplo, si ocultamos 54 bits (un coordenada GPS: 42.08.36 → 42º 08' 36'') utilizando como fuente de entrenamiento la versión digital de "100 años de soledad" y un orden 3, se obtiene un estegotexto como el de la Figura 13 (hay 17.070 posibles):

Figura 13. Ejemplo de estegotexto generado automáticamente en lengua española. [stegotext-540]. Ocultación 54 bits. Texto fuente versión digital de "100 años de Soledad" de Gabriel García Márquez. 2,5714 bits/palabra. Tamaño fuente: 809KB texto plano, 137.649 palabras. Orden de complejidad 3. Expansión 1:14

> método de exterminio, desde entonces hasta la hora de que se le dio la muerte del mundo. El coronel Aureliano Buendía

Este estegotexto podría ser corregido manualmente y generar un estegotexto con apariencia próxima a uno generado por un ser humano (Figura 14, Figura 15, Figura 16).

Figura 14. Corrección manual del estegotexto de la Figura13, mejorando cohesión y coherencia global. Ocultación de 54 bits. Capacidad de ocultación final 1,5 bits/palabra

> método de exterminio de esclavos, desde entonces hasta la hora de su sublevación. Muerte que se le dio sin piedad, la muerte del mundo menos afortunado. El coronel Aureliano Buendía pagaría años después por sus atrocidades.

Figura 15. Orden n=3, 126 bits ocultos, texto de entrenamiento "Poesías Completas. Antonio Machado" [ST-180]

> La tarde se ha ido llegando las hojas de la fuente se oía tañer de una tierra. Nunca se cansa. Pasado habían el agua muda que enorme muro de la fuente. Yo no conozco el agrio zumo dorado de amor. El tren, abril galán. ¡Oh, dime si son mías. La tarde caía, qué a mí

Figura 16. Ejemplo de maquillaje manual del estegotexto de la Figura 15. Capacidad de ocultación 1 bit palabra. En negrita las palabras originales

> **La tarde se ha ido llegando** y el viento contra **las hojas** esculpidas **de la fuente se oía tañer de** campanillas, melodía para **una tierra** entristecida. Esa fuente de la que fluye agua como néctar sin cesar. **Nunca se cansa**. **Pasado** el tiempo no **habían** regresado, enamorados que bebieran **el agua muda** de la fuente **que** en otra época, tras un **enorme muro**, retaba a demostrar su amor bebiendo libertad **de la fuente** prohibida**. Yo no conozco el agrio** muro y tampoco el **zumo dorado de amor** de esos jóvenes**. El tren** del amor ya pasó**,** mi **abril** cuando fui **galán** pasó como estrella fugaz en el cielo. **¡Oh, dime si** puedo recuperarlo! Mis paranoias **son mías** pero puede que haya esperanza**. La tarde caía, qué** rápido... **a mí** me pareció como un suspiro.

ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

4. CONCLUSIONES

En este trabajo se demuestra cómo es posible generar automáticamente estegotextos en lengua española mediante un modelo N-Gram. Se observa cómo imitar estadísticamente uno o más textos de entrenamiento a nivel de palabra permite una imitación léxica y gramatical mejor que propuestas previas, como por ejemplo el algoritmo de Wayner. No obstante, la utilización de un modelo N-Gram sigue produciendo estegotextos carentes de coherencia y de sentido global (especialmente si son largos). Mientras nuevas investigaciones permiten solucionar este hecho, se propone el concepto de maquillaje manual. Es posible editar estegotextos generados para corregir errores y darle una "apariencia humana" al resultado. Las pruebas realizadas indican que es factible obtener estegotextos indistinguibles por un lector humano y entendemos que si un humano no puede determinar si un texto ha sido escrito por un humano o una máquina, un software automatizado no será capaz de hacerlo tampoco. Las pruebas realizadas indican que es factible ocultar unas decenas o pocas centenas de bits en textos de decenas o pocas centenas de palabras y demuestran una ocultación mínima de 0,5 bits por palabra en el estegotexto final. Si se opta por el maquillaje manual de estegotextos que oculten decenas de bits, se observa la necesidad de invertir pocos minutos en la corrección, lógicamente esta depende de la capacidad de edición del emisor.

Esta aparentemente baja capacidad de ocultación permite, al menos, enmascarar direcciones de internet o mensajes breves (como podrían ser mensajes de movilización, localización, etc.). Por ejemplo, en una red monitorizada es posible intercambiar un "texto" inofensivo que contenga la posición GPS de una reunión clandestina o es posible intercambiar "textos" en una red corporativa que sirvan para controlar un software espía de manera transparente a los filtros de la red, etc.

Actualmente las investigaciones en curso se orientan hacia cuatro aspectos: La automatización y simplificación del maquillaje manual; aumentar la capacidad de ocultación, la introducción de modelos lingüísticos para facilitar estegotextos con mayor coherencia global y un análisis en profundidad de la seguridad de la propuesta. Destáquese que actualmente emisor y receptor de una comunicación deben conocer previamente el texto de entrenamiento utilizado. Un atacante podría intentar descubrir el texto de entrenamiento a partir de fragmentos capturados de estegotextos transmitidos. La tarea no es baladí y es exclusiva por estegotexto capturado. El orden de entrenamiento utilizado y el maquillaje manual empleado introducen la posibilidad de cuantiosos falsos positivos si se desea buscar textos que cumplan ciertos patrones N-GRAM. La estrategia a seguir de ataque no es sencilla de definir dado que no basta con hacer búsquedas en buscadores de internet u otros recursos y realizar comparaciones masivas. Este tema está abierto a la investigación.

NOTAS

1 La ciencia de la esteganografía puede definirse como la ciencia y el arte de ocultar una información dentro de otra, que haría la función de tapadera o cubierta, con la intención de que no se perciba ni siquiera la existencia de dicha información (Kahn, 1996; Cox, 2008). La ciencia de la esteganografía es complementaria a la ciencia de la criptografía; esta última si bien no oculta la existencia de un mensaje, sí lo hace ilegible para quien no esté al tanto de un determinado secreto, una clave. En la práctica ambas ciencias pueden combinarse para mejorar la autenticidad y privacidad de las comunicaciones.

2 ¿Es posible que una máquina genere textos en lenguaje natural que un humano no sepa distinguir si los ha producido una máquina o una persona?

ARBOR Vol. 190-768, julio-agosto 2014, a160. ISSN-L: 0210-1963 doi: http://dx.doi.org/10.3989/arbor.2014.768n4014

BIBLIOGRAFÍA

Bergmair, R. (2007). A comprehensive bibliography of linguistic steganography. *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*. http://dx.doi.org/10.1117/12.711325

Blasco, J., Hernández-Castro, J., Tapiador, J. y Ribagorda, A. (2008). Csteg: Talking in C code. *Proceedings of SECRYPT International Conference*, pp. 399–406.

Chand, V. y Orgun, C. (2006). Exploiting linguistic features in lexical steganography: design and proofof-concept implementation. *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06)*, 6, p. 126b. http://dx.doi.org/10.1109/HICSS.2006.175

Chapman, M. y Davida, G. (1997). Hiding the hidden: A software system for concealing ciphertext as innocuous text. *Proceedings of the International Conference on Information and Communication Security*. *Lecture Notes in Computer Sciences*, 1334.

Chapman, M., Davida, G. y Rennhard, M. (2001). A practical and effective approach to large-scale automated linguistic steganography. *ISC '01 Proceedings of the 4th International Conference on Information Security*, pp. 156-165.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Cox, I., Miller, M., Bloom, J., Fridrich, J. y Kalker, T. (2007). *Digital Watermarking and Steganography* (2ª ed.). San Francisco: Morgan Kaufmann Publishers.

Dai, W., Yu, Y. y Deng, B. (2009). BinText steganography based on Markov state transferring probability. *ICIS '09 Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, New York: ACM, pp. 1306-1311.

Grothoff, C., Grothoff, K., Alkhutova, L., Stutsman, R. y Atallah, M. (2005). Translation-Based Steganography. *Computer Science Information Hiding. Lecture Notes in Computer Science*, 3727, pp. 219-233. http://dx.doi.org/10.1007/11558859_17

Kahn, D. (1996). *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet.* New York: Scribner.

Kerckhoffs, A. (1883). La cryptographie militaire. *Journal des sciences militaires*, IX, pp. 5-38 y 161-191.

Meng, P., Hang, L., Yang, W., Chen, Z. y Zheng, H. (2009). Linguistic Steganography Detection Algorithm Using Statistical Language Model. *International Conference on Information Technology and Computer Science 2009 (ITCS 2009)*, pp. 540-543. http://dx.doi.org/10.1109/ITCS.2009.246

Meng, P., Liusheng, H., Zhili, C., Yuchong, H. y Yang, W. (2010). STBS: A Statistical Algorithm for Steganalysis of Translation-Based Steganography. *Lecture Notes in Computer Science*, 6387, pp. 208-220. http://dx.doi.org/10.1007/978-3-642-16435-4_16

Muñoz, A., Argüelles, I. y Carracedo, J. (2009). Modificaciones sintácticas en lengua española con utilidad en esteganografía lingüística. *Revista Electrónica de Lingüística Aplicada*, 8, pp. 229-247.

Muñoz, A. y Argüelles, I. (2012). Modificaciones sintácticas basadas en la reordenación de complementos del verbo con utilidad en esteganografía lingüística. *Revista Electrónica de Lingüística Aplicada*, 10, pp. 31-54.

Shu-feng, W. y Huang, L. (2003). *Research on Information Hiding*. Degree of master, University of Science and Technology of China.

Topkara, M. Topkara, U. y Atallah, M. (2007). Information Hiding through Errors: A Confusing Approach. *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*, 29. http://dx.doi.org/10.1117/12.706980

Wayner, P. (1992). Mimic functions. *Cryptologia*, XVI, pp. 193–214. http://dx.doi.org/10.1080/0161-119291866883

Wayner, P. (1995). Strong theoretical steganography. *Cryptologia*, XIX, pp. 285–299. http://dx.doi.org/10.1080/0161-119591883962

Zuxu, D., Fan, H., Muxiang, Y. y Guohua, C. (2007). Text Information Hiding Based on Part of Speech Grammar. *Proceedings of the 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007)*, pp. 632-635. http://dx.doi.org/10.1109/CISW.2007.4425575