Revista CIENCIA Y TECNOLOGÍA V.16 N.1

ISSN 1810-6781 Rev. Cienc. Tecnol. 16(1): 33-42, (2020)

Procesamiento del Lenguaje Natural para Reconocer Mensajes de Textos Extorsivos a través del Análisis Sintáctico y Lematización

Natural Language Processing to Recognize Extortive Text Messages through Syntactic Analysis and Lemming

Juan Carlos Obando Roldán[1,*]; José Arturo Pulido Díaz [2]; José Alberto Gómez Ávila[1]

[1] Escuela de Ingeniería de Sistemas - Facultad de Ingeniería, Universidad Nacional de Trujillo, Av. Juan Pablo II s/n – Ciudad Universitaria, Trujillo, Perú.
[2] Escuela de Informática – Facultad de Ciencias Físicas y Matemáticas, Universidad Nacional de Trujillo, Av. Juan Pablo II s/n – Ciudad Universitaria, Trujillo, Perú.

* Autor correspondiente: docenteunt@yahoo.es (J. Obando)

RESUMEN

Este trabajo de investigación muestra como a través del análisis sintáctico y lematización se puede procesar el lenguaje natural utilizado por los seres humanos para comunicarse y determinar si el contenido de un mensaje de texto es o no extorsivo. Los mensajes de textos extorsivos representan un grave daño a la libertad de las personas y contribuyen al clima de inseguridad ciudadana. El uso de análisis sintáctico y lematizado permitió la formulación de reglas de inferencia que formalizaron el conocimiento del experto policial y que fueron implementadas en lenguaje de programación Prolog[1] y para probar su funcionamiento se utilizó interfaces de apoyo basadas en lenguaje de programación Java[2]. La implementación permitió concluir el tratamiento de la información para deducir el nivel o el grado de malicia en los mensajes de textos extorsivos.

Palabras clave: extorsión; mensajes de texto; análisis sintáctico; lematización.

ABSTRACT

This research work shows how through the syntactic analysis and stemming the natural language used by human beings can be processed to communicate and determine whether the content of a text message is extortive or not. Texts of extortive texts represent a serious damage to the freedom of the people and contribute to the climate of citizen insecurity. The use of syntactic and lematized analysis allowed the formulation of inference rules that formalized the knowledge of the police expert and were implemented in the Prolog programming language and to test its operation, it was used support interfaces based on the Java programming language. The implementation allowed to conclude the treatment of the information to deduce the level or degree of malice in the messages of extorsive texts.

Keywords: extortion; text messages; syntactic analysis; lemmatized.

1. INTRODUCCIÓN

La naturaleza neutra de los avances tecnológicos en telefonía y comunicaciones electrónicas no ha impedido que su uso pueda derivarse hacia actos delictivos realizados por organizaciones criminales. Según el informe del Observatorio Nacional de Seguridad Ciudadana (Mininter, 2018) en nuestro país autoridades, empresarios, profesionales entre otros han sido víctimas de la extorsión usando llamadas o mensajes de texto maliciosos. Según el informe anteriormente mencionado en el 2016, las víctimas por delitos de extorsión llegaron a representar el 0,28%, lo que significa aproximadamente 51 700 víctimas. Sólo el 10% de las víctimas denunció el hecho, es decir 5 170. Esta información da a entender que los delitos de extorsión se encuentran subregistrados, lo que favorece a que se perpetúe la realización del delito y dificulta la implementación de políticas públicas orientadas a combatirlos.

[1] Lenguaje de programación basado en el paradigma de la programación lógica.
[2] Lenguaje de programación basado en el paradigma de la programación orientada a objetos.

Autor Responsable: Juan Carlos Obando Roldán Fecha de envío: 19-07-2019 Fecha de aprobación: 08-01-2020

La extorsión es un hecho punible consistente en obligar a una persona, a través de la utilización de violencia o intimidación a realizar u omitir un acto o negocio jurídico con ánimo de lucro y con la intención de producir un perjuicio de carácter patrimonial o bien. (Medina, 2015). Extorsión también significa la presión que, mediante amenazas, se ejerce sobre alguien para obligarle a obrar en determinado sentido.

Si tomamos en cuenta la importancia del lenguaje como un factor primordial para un acercamiento inicial a la naturaleza de las conductas sociales, es significativo que la raíz misma de este concepto se refiera al ejercicio de actos violentos perjudiciales en que la voluntad individual de la persona afectada se ve prácticamente anulada y uno de esos medios han sido las llamadas telefónicas y los mensajes de texto.

La utilización de mensajes de texto basados en lenguaje natural que es el lenguaje utilizado por los seres humanos para comunicarse y que ha evolucionado con el tiempo para fines de comunicación humana, como el español o alemán puede ser sometido a procesamiento (Brookshear, 2015). El procesamiento del lenguaje natural está compuesto por la comprensión y la generación. La comprensión se inicia a partir de una frase escrita o hablada y consiste en obtener una representación formal que permita posteriormente efectuar las acciones adecuadas a la información recibida. La generación consiste en transformar una representación formal de algo que se quiere comunicar, a una expresión en algún lenguaje natural, escrito o hablado.

Son fases en el proceso de comprensión del lenguaje natural la percepción del reconocimiento del habla y/o escritura como primera fase, como segunda fase el análisis sintáctico que consiste en obtener la estructura de una frase a partir de la secuencia de palabras y el análisis semántico, que consiste en obtener un significado a partir de la estructura sintáctica así como la eliminación de ambigüedades y como tercera fase el escoger uno de los posibles significados e incorporación a la base de conocimiento.

A esto si sabemos que la gramática (sintaxis) de un lenguaje en general puede expresarse como un conjunto de producciones, que definen de manera declarativa al lenguaje, es decir, que determinan las cadenas de símbolos correctas o sentencias que forman ese lenguaje (Reyes et al. 2013). Estas reglas pueden expresarse en la sintaxis del lenguaje de programación Prolog que es un lenguaje de programación basado en el paradigma de la programación lógica, y acompañadas de los hechos que definen los símbolos terminales y de consultas, constituyen programas cuya ejecución conduce a la comprobación o la formación de sentencias. De hecho, el lenguaje de programación Prolog surgió de investigaciones sobre procesamiento del lenguaje natural, y éste sigue siendo uno de sus campos de aplicación.

Consideremos la lengua española para describir la formación de sentencias (oraciones) utilizamos categorías sintácticas o sintagmas, como «nombre», «verbo», etc. (o «sintagma nominal», «sintagma verbal», etc.). Está claro que, si bien «nombre» es un nombre, «verbo» no es un verbo. Esto es debido a que para describir el lenguaje utilizaremos como metalenguaje el propio español, y ponemos las comillas para destacar que la palabra en cuestión se utiliza como elemento del metalenguaje. La notación habitualmente seguida para indicar que se trata de un elemento del metalenguaje consiste en encerrar la palabra entre paréntesis angulares: <nombre>, <verbo>, etc.

Aunado al análisis sintáctico también tenemos a la lematización, proceso lingüístico que consiste en dada una forma flexionada, hallar el lema correspondiente (Gómez, 2005). El lema es la palabra que se acepta como palabra base de todas las formas flexionadas de una misma palabra. Por ejemplo, el lema de casas es casa, carro es el lema de carros, carritos. La lematización se puede implementar a través del desarrollo de programas de análisis morfológico. Hay diversos grados de lematización posible: podemos hacer una lematización puramente morfológica, o bien hacer una lematización sintáctica que tenga en cuenta el contexto en el que aparece la palabra. Por ejemplo, en un análisis morfológico la palabra ama tendría dos lemas: el sustantivo ama y el verbo amar. Sin embargo, en un contexto sintáctico, podemos desambiguarlo y optar por un único lema.

Un lematizador es un programa capaz de obtener los lemas de una palabra. El lema es la raíz principal de una familia morfológica correspondiente. Por ejemplo, en el caso de la palabra "perritas", el lematizador determina que se trata del lema "perro" con los atributos femenino, plural y diminutivo. De manera similar, si ingresamos la palabra "leerá", nos indicará que es la tercera persona del singular del futuro del indicativo del verbo "leer". El lematizador deberá estar en capacidad de encontrar todos los posibles lemas; así, si se busca la palabra "suma", el lematizador informa lo siguiente:

- Raíz del sustantivo "suma".
- Tercera persona singular del presente de indicativo del verbo "sumar".
- Tercera persona singular del imperativo del verbo "sumir".
- Femenino singular del adjetivo "sumo".

2. MATERIALES Y MÉTODOS

El procedimiento para la verificación del contenido extorsivo del mensaje de texto ha seguido el siguiente procedimiento. En primer lugar realizar el análisis sintáctico del mensaje de texto basado en lenguaje natural, luego proceder al análisis de reglas, seguidamente a través del uso de la lógica proposicional formalizar el conocimiento a través de reglas de inferencia. Posteriormente se procede a elaborar el diseño de entradas así como la formulación de reglas que apoyados por una red semántica y los lenguajes de programación Prolog y Java que implementan el lematizado de palabras permitieron hacer las pruebas y apreciar los resultados.

2.1 Análisis Sintáctico

Una sentencia castellana sintácticamente correcta estará compuesta por palabras concretas pertenecientes a las diversas categorías sintácticas (uno o varios sintagmas nominales, uno o varios sintagmas predicativos, etc.) combinadas de acuerdo con ciertas reglas. Usando la figura 1 podemos formular algunas reglas.

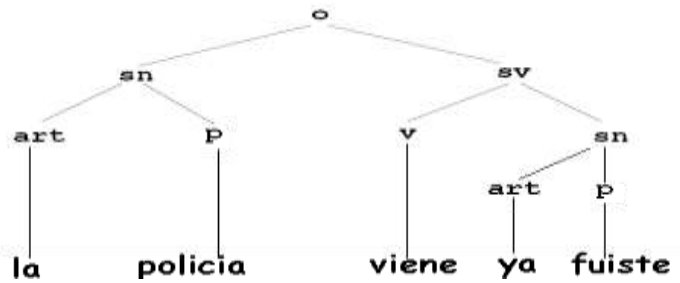

Figura 1. Árbol sintáctico.

Por ejemplo, una regla puede ser:

- Para formar una sentencia, póngase un sintagma nominal y a continuación un sintagma predicativo.

Y otras:

- Un nombre propio es un sintagma nominal.
- «Policía» es un nombre propio.
- Un sintagma predicativo puede formarse con una forma verbal sustantiva y un sintagma nominal.
- «ya» es una forma verbal sustantiva.
- Un sintagma nominal puede formarse con un determinante y un nombre común.
- Un artículo es un determinante.
- «la» es un artículo.
- «fuiste» es un nombre común.

Del conjunto de estas nueve reglas puede deducirse que «la Policía viene ya fuiste» es una sentencia (u «oración») del castellano. Las reglas anteriores pueden expresarse formalmente (utilizando ya el convenio de expresar los elementos del metalenguaje mediante «<...>») así:

- <S> → <SN> <SP>
- <SN>→<Npr>
- <Npr> → Policia
- <SP> → <FVs> <SN>
- <FVs> → viene
- <SN> → <Det> <Ncom>
- <Det> → <Art>
- <Art> → la
- <Ncom> → estado

(El símbolo «→» es equivalente al «::=» de la notación Backus-Naur o BNF).

La oración obtenida puede descomponerse sintácticamente de acuerdo con las reglas; esta descomposición puede indicarse mediante corchetes etiquetados:

[[[policía]] [[viene] [[[ya]] [fuiste]]]]

S SN Npr SP FVs SN Det Art Ncom

o bien, de forma gráfica, con un árbol llamado árbol de derivación o árbol sintáctico (Figura 1)

Las frases (o sentencias) del español pueden considerarse como listas de palabras, y así, por ejemplo, «la policía viene ya fuiste» se representará como «[’Policía’,viene,ya,fuiste]» . A cada categoría sintáctica (sentencia, artículo, etc.) le corresponderá un predicado: «sentencia(S)» será verdadero si la lista S es una sentencia correcta («S» es una variable con nombre arbitrario; igual podría llamarse «X»).

Las reglas de escritura terminales (aquellas que tienen en su parte derecha palabras del lenguaje), como «<Art> → la» , se representarán como hechos: «articulo([la])» , y las auxiliares (aquellas en cuya parte derecha figura algún elemento del metalenguaje), por ejemplo,

<SP> → <FVs><SN>

como reglas Prolog que expresen el significado de la regla; en ese ejemplo, «un sintagma predicativo se forma concatenando una forma verbal y un sintagma nominal» , es decir:

sint_pred(SP) :- forma_v_s(FVS),sint_nom(SN), concatena(FVT,SN,SP).

(De nuevo, «SP» , «FVS» y «SN» son nombres arbitrarios de variables).

De acuerdo con esto, y escribiendo los hechos y reglas Prolog en el mismo orden en que aparecen las reglas de escritura dadas anteriormente, resulta:

```
sentencia(S):-sint_nom(SN), sint_pred(SP),
        concatena(SN,SP,S).
sint_nom(SN):-nom_prop(SN).
        nom_prop([’Espana’]).
        sint_pred(SP):-forma_v_s(FVS),
        sint_nom(SN),
        concatena(FVS,SN,SP).
forma_v_s([es]).
sint_nom(SN):-det(DET),
nom_com(NC),
concatena(DET,NC,SN).
det(X):-art(X).
art([un]).
nom_com([estado]).
 concatena([],L,L).
concatena([X|L1],L2,[X|L3]) :- concatena(L1,L2,L3).
```

Ante la consulta: ?- sentencia(X).

El sistema encontrará todas las sentencias que pueden formarse de acuerdo con las reglas dadas:

X=[’Policia’,viene,’Policia’]
X=[’Policia’,viene,ya,fuiste]
X=[ya,fuiste,viene,’Policia’]
X=[ya,fuiste,viene,ya,fuiste]

Entonces cómo se puede entender se consigue una interacción mas natural con la computadora (es decir que, aunque la representación interna sea en forma de listas, la salida del computador y, eventualmente, la entrada sean cadenas de palabras tal como las escribimos habitualmente).

2.2 Lematización

El método de la lematización es empleado en este trabajo, luego de la depuración de las frases en palabras que sucede a través del análisis sintáctico. Específicamente se a utilizado el método de eliminación de afijos. Este método elimina afijos y/o sufijos de los términos para conseguir un lema. Para la eliminación de afijos se a empleado el algoritmo de Porter el que consiste en tres reglas de condición/acción.

Condiciones en el lema

- La medida m de un lema se basa en sus alternaciones de secuencias vocal-consonante → [C](VC)m[V]
- El lema termina con una letra X dada → *<X>
- El lema contiene una vocal → *v*
- El lema termina en una doble consonante → *d
- El lema termina con una secuencia consonante-vocal-consonante, donde la última consonante no es w, x ni y → *o

Condiciones en el sufijo: Las condiciones del sufijo toman la forma: (sufijo_actual == patrón)

Condiciones en las reglas: Las reglas se dividen en pasos. Las reglas en un paso son examinadas secuencialmente, y sólo una de estas puede ser aplicable.

2.3 Análisis de Reglas

Para almacenar todo el conocimiento obtenido, en la base de conocimiento se hace uso de las reglas. Estas reglas relacionan dos o más afirmaciones para determinar la creencia en las conclusiones, en nuestro caso las proposiciones corresponden a los mensajes que el ciudadano (persona buena) recibe del extorsionador (persona mala) y las conclusiones hacen referencia al tipo de amenaza presente y su respectivo tratamiento.

A continuación, se tienen las reglas para el diagnóstico y tratamiento de los mensajes en el ciudadano.

Regla 1.

mensaje (palabra ofensiva, significado preventivo).

Tratamiento: Comparación con palabras de la base de datos de conocimiento lingüístico. Si hay más de 2 palabras ofensivas entonces significado es igual a amenaza.

Regla 2.

mensaje (palabra mal escrita, significado preventivo).

Tratamiento: Lematizar palabra con la base de datos de conocimiento lingüístico, profundizar en el origen del significado de la palabra.

Regla 3.

Diagnóstico de llamada (número telefónico, normal o denunciado).

Tratamiento: comparar número telefónico en lista de números denunciados por cualquier tipo de delito común entonces Bloquear llamada y enviar mensaje de voz, al extorsionador que su número está bloqueado / denunciado para hacer llamadas.

2.4 Reglas de Inferencia

Para ejemplificar a través de reglas de inferencia, se formalizará el conocimiento haciendo uso de la lógica proposicional y de esta forma mostrar cómo se puede inferir en base a cierto conocimiento y se mostrará algunas de sus reglas.

Regla 1

P1: palabras con significado amenazante.
P2: palabras con significado ofensivo.
P3: si el mensaje contiene palabras ofensivas y palabras amenazantes entonces mensaje malicioso.
Formalizando:
p: palabras amenazantes, q: palabras ofensivas
r: mensaje malicioso
P1: p
P2: q
P3: $p \wedge q \rightarrow r$

P4: $p \wedge q$	Ley de conjunción [P1, P2]
P5: r	Modus Ponendo Ponens [P3, P4]

Regla 2

P1: palabras comunes con falta de ortografía.
P2: raíz gramatical de la palabra con significado común.
P3: si el mensaje presenta palabras comunes con falta de ortografía, pero con significado común entonces mensaje malicioso
Formalizando:
p: palabras mal escritas q: raíz gramatical tiene significado común
r: mensaje malicioso
P1: p
P2: q
P3: $p \wedge q \rightarrow r$

P4: $p \wedge q$	Ley de conjunción [P1, P2]
P5: r	Modus Ponendo Ponens [P3, P4]

Regla 3

P1: número telefónico común.
P2: número telefónico denunciado por extorsión o delito.
P3: llamada bloqueada.

P4: número telefónico detectado como numero delictivo.
Formalizando:
p: número telefónico normal
q: número telefónico denunciado
s: bloqueo de llamada
t: numero detectado como malicioso
P1: p
P2: q
P3: s
P4: p ∧ q → s
P5: s → t

P6: p ∧ q	Ley de conjunción [P1, P2]
P7: s	Modus Ponendo Ponens [P4, P6]
P8: t	Modus Ponendo Ponens [P4, P7]

2.5 Diseño de Entradas

El sistema inteligente como entrada tendrá los mensajes para el desarrollo del sistema, para el cual se realizan preguntas cuyas respuestas son SI o No; también pueden responder con las opciones como ser: advertencia (leve, moderada), amenaza (grave), bloqueo, las cuales se verán reflejadas en la tabla 1.

Tabla 1. Descripción de variables lingüísticas

Variables	Nombre de los síntomas	Variables lingüísticas
S1	significado	Advertencia leve, moderada,
S2	llamada	amenaza, bloqueo
S3	falta de ortografía	Si, No
S4	amenazante	Si, No

2.6 Formulación de Reglas

Regla 1: Si el mensaje contiene palabras amenazantes y palabras ofensivas entonces se trata de un mensaje malicioso

Por lo tanto:
P: palabras amenazantes,
Q: palabras ofensivas
M: mensaje malicioso
Fórmula: ∀x (Mx → (Px ∧ Qx))

Regla 2: Algunos mensajes contienen palabras mal escritas cuya raíz gramatical tiene un significado común entonces se trata de un mensaje malicioso

Por lo tanto:
P: palabras mal escritas,
Q: raíz gramatical tiene significado común
M: mensaje malicioso
Fórmula: ∀x ∃y (Mx →[P(x,y) ∧ Q(y)])

Regla 3: Cualquier llamada telefónica que proviene de un número telefónico común, no es una llamada extorsiva.

Algunas llamadas telefónicas que provengan de un número telefónico denunciado por extorsión o cualquier acto delictivo la llamada es bloqueada.
Por lo tanto:
P: número telefónico normal
Q: número telefónico denunciado
B: bloqueo de llamada
Formula:

a. ∀x (Px → -Bx)
b. ∃x (Qx → Bx)

3. RESULTADOS Y DISCUSION

Para probar el funcionamiento de las reglas usamos una red semántica a nivel del significado de un mensaje de texto basado en lenguaje natural. La figura 2 muestra la red semántica a utilizar.

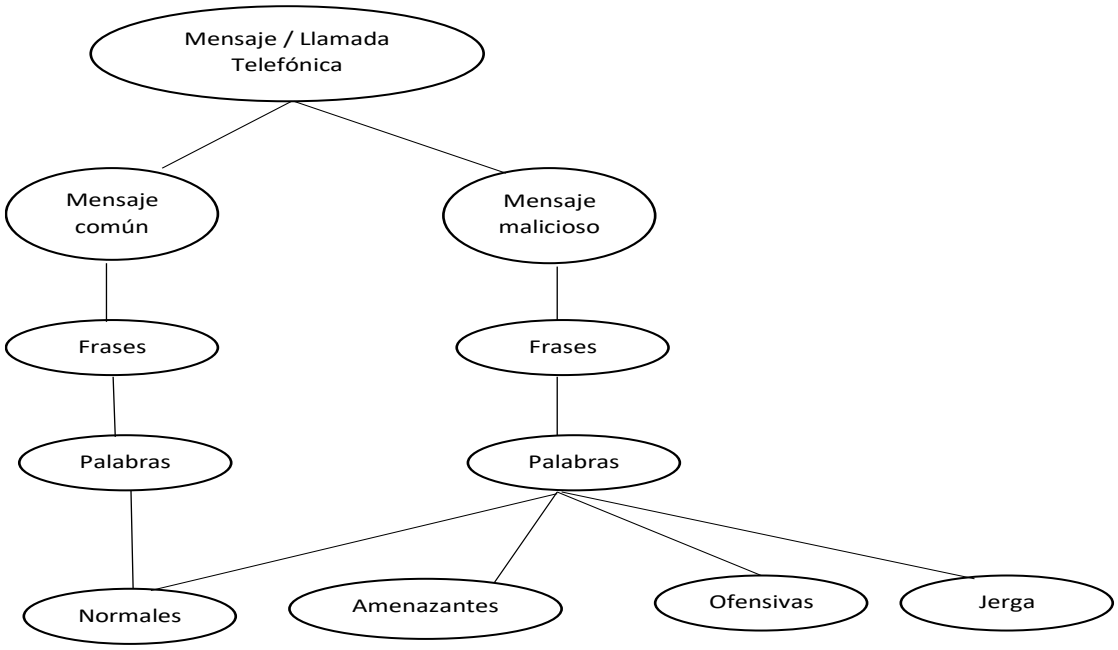

Figura 2. Árbol de red semántica

Este árbol semántico se implementa posteriormente en el lenguaje de programación Prolog y se desarrollan las interfaces de apoyo basadas en el lenguaje de programación Java. En las siguientes páginas se muestran los códigos de implementación de la base de conocimientos así como el shell del sistema experto.

Base de Conocimientos

```
:- dynamic conocimiento/2.
conocimiento(muerte,
        ['el usuario es amenazado que morirá dentro de poco',
        'el usuario teme por su vida']).
conocimiento(matar,
        ['el usuario es amenazado de muerte',
        'el usuario teme por su vida']).
conocimiento(dinero,
        ['el usuario es amenazado sino cumple con el dinero',
        'al usuario se le exige una suma de dinero']).
conocimiento('dar de baja',
        ['el criminal está decidido matar',
        'condiciona a su víctima sino la matara']).
conocimiento('Caerse el caset',
        ['Hablar más de la cuenta.',
        'Dar información a la policía']).
conocimiento('Cara de callo',
        ['Caradura, persona que no se inmuta frente a una conducta delictual, aun cuando sea descubierto']).
conocimiento('Cuchillos largos',
        ['presos de extrema violencia']).
conocimiento('Cuetazo',
        ['disparo de arma de fuego']).
```

Sistema Experto: experto.pl

```
/*Trata los mensajes como una lista de palabras
Utiliza asser t/1 para cambiar dinámicamente la base de conocimientos.
Determina la verdad y falsedad de las palabras conocidas. Puede contestar a las preguntas 'porque' e incluye
capacidad de explicación. Elimina dinámicamente las aseveraciones agregadas después de cada consulta.*/
:- dynamic conocido/1.
consulta:-
        haz_comprobacion(X),
        escribe_comprobacion(X),
        ofrece_explicacion_comprobacion(X),
        clean_scratchpad.
consulta:-
        write('No hay suficiente conocimiento para elaborar una Comprobacion.'),
        clean_scratchpad.
haz_comprobacion(Comprueba):-
        obten_hipotesis(Comprueba, ListaDePalabras),
        prueba_presencia_de(Comprueba, ListaDePalabras).
obten_hipotesis(Comprueba, ListaDePalabras):-
        conocimiento(Comprueba,ListaDePalabras).
prueba_presencia_de(Comprueba, []).
prueba_presencia_de(Comprueba, [Head | Tail]):-
        prueba_verdad_de(Comprueba, Head),
        prueba_presencia_de(Comprueba, Tail).
prueba_verdad_de(Comprueba, Palabra):-
        conocido(Palabra).
prueba_verdad_de(Comprueba, Palabra):-
        not(conocido(is_false(Palabra))),
        pregunta_sobre(Comprueba,Palabra,Reply),Reply=si.
pregunta_sobre(Comprueba, Palabra, Reply ):-
        write('Es verdad que '),
        write(Palabra),write('? '),
        read(Respuesta), process(Comprueba,Palabra, Respuesta,Reply ).
process(Comprueba,Palabra,si,si):-
        asserta(conocido(Palabra)).
process(Comprueba,Palabra,no,no):-
        asserta(conocido(is_false(Palabra))).
process(Comprueba,Palabra,porque,Reply):- nl,
        write('Estoy investigando la hipotesis siguiente: '),
        write(Comprueba), write( '.'), nl,write('Para esto necesito saber si'),
        write(Palabra),write('.'), nl, pregunta_sobre(Comprueba, Palabra, Reply ).
process(Comprueba,Palabra,Respuesta,Reply):-
        Respuesta \== no,
        Respuesta \== si,
        Respuesta \== porque, nl,
        write('Debes contestar si, no o porque.'), nl,
        pregunta_sobre(Comprueba,Palabra, Reply) .
escribe_comprobacion(Comprueba):-
        write('La Comprobacion es '),
        write(Comprueba), write( '.'), nl.
ofrece_explicacion_comprobacion(Comprueba):-
        pregunta_si_necesita_explicacion(Respuesta),
        actua_consecuentemente(Comprueba,Respuesta).
pregunta_si_necesita_explicacion(Respuesta):-
        write('Quieres que justifique esta Comprobacion? '),
        read(RespuestaUsuario),
        asegura_respuesta_si_o_no(RespuestaUsuario,Respuesta).
asegura_respuesta_si_o_no(si,si).
```

La figura 3 expone la interfaz desarrollada en lenguaje de programación Java, que captura el número y el contenido del mensaje de texto.

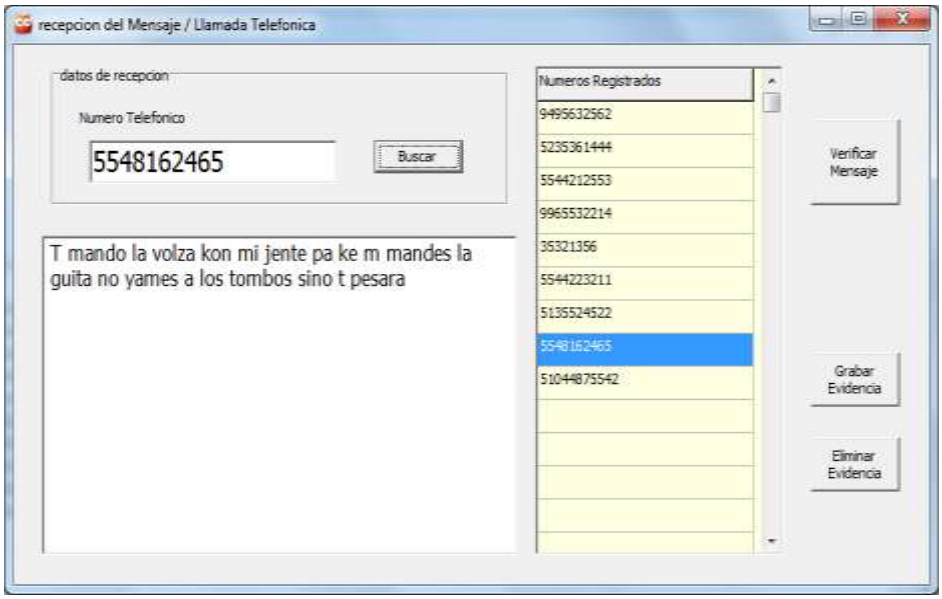

Figura 3. Interfaz de captura de mensaje de texto.

Haciendo clic en el botón "Verifica mensaje", nos muestra como resultados en Prolog, la secuencia e identificación para clasificar si el presente mensaje es Extorsivo, o no.

4. CONCLUSIONES

Se validó el uso del análisis sintáctico y del lematizado para el reconocimiento de si un mensaje de texto es o no extorsivo, analizando la estructura clausular desde una perspectiva pragmática considerando los patrones del habla adoptados por un estrato social delictivo cuando teledirige mensajes de texto a sus víctimas.

Esta investigación representa una aportación significativa y original al análisis sintáctico de los lenguajes de adjunción de árboles y por extensión al procesamiento del lenguaje natural a través del análisis sintáctico empleando como recurso la lematización, mediante la cual las palabras de un texto que pertenecen a un mismo paradigma flexivo o derivativo son llevadas a una forma normal que representa a toda la clase. Nos referimos a la normalización, como la forma de separar las palabras analizadas en un núcleo conceptual (lexema) y agregados morfológicos (morfemas). En este caso, la lematización se usó para encontrar el lexema de las palabras analizadas, eliminando sufijos morfológicos (stemming en la literatura técnica).

Se ha demostrado que es posible establecer un camino evolutivo continuo en el que se sitúan los algoritmos de análisis sintáctico que incorporan las estrategias de análisis más importantes, basándose en esquemas de lematización, adaptando conclusiones con reglas de inferencia para determinar a través de la lógica fuzzy una solución más óptima que se sitúa en los resultados que origina un sistema experto.

Por todo eso es que dentro del marco de ejecución del Plan de Desarrollo Regional Concertado de la Región La Libertad que en su dimensión social señala como objetivo: "Impulsar la seguridad ciudadana, el respeto a las leyes y la convivencia social" está investigación contribuye en el control de mensajes de texto que vienen siendo utilizados como medios de extorsión.

AGRADECIMIENTOS

Un agradecimiento al personal del Vicerrectorado de Investigación a través de su Oficina de Investigación por el apoyo constante en el desarrollo de la presente investigación. Así mismo al alto mando de la Policía Nacional del Perú por brindarnos su experticia en el tema de las extorsiones. También el agradecimiento va a todos los docentes de las escuelas de Ingeniería de Sistemas e Informática que gracias a sus apreciaciones y comentarios permitieron la culminación del presente trabajo.

REFERENCIAS BIBLIOGRÁFICAS

Brookshear J. Glean. 2015. Teoría de la computación: lenguajes formales, autómatas y complejidad. 1era Edición. Editorial Addison Wesley Longman. Wilmington, USA. 465 pp.

Gómez Díaz, R. 2005. La lematización en español: una aplicación para la recuperación de información. 1era Edición. Editorial Trea. Gijón, España. 240 pp.

Medina, A. 2015. Las bandas de extorsión y sus causas determinantes en Florencia de Mora Trujillo. Tesis de Licenciatura. 14: 23-25.

Mininter 2018. Cifras de la Extrosión en el Perú. Observatorio Nacional de Seguridad Ciudadana. Ministerio del Interior del Perú. Disponible en https://observatorio.mininter.gob.pe/reportes/cifras-de-extorsion-en-el-peru

Reyes, J; Montes, A; Gonzales, J; Pinto, D 2013 Clasificacion de roles semanticos usando caracteristicas sintacticas, semanticas y contextuales. Jornadas de Computacion y Sistemas 321:263-272.